View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

ELSEVIER
Theoretical Computer Science 289 (2002) 1–49
Theoretical Computer Science
www.elsevier.com/locate/tcs

A generic causal model for place latency

Reiner Durchholz

GMD – German National Research Center for Information Technology, Institute for Algorithms and Scientific Computing, D-53754 Sankt Augustin, Germany

Received October 1999; received in revised form July 2001; accepted August 2001
Communicated by G. Rozenberg

Abstract

For a prototypical class of time-extended Petri nets it is shown that the extension does not increase their expressive power. The nets in this class have token latencies attributed to places. Place latency nets are formally defined together with their firing semantics. For any place latency system, an explicit construction of an elementary net system is given as an implementation, which is proven to be behaviourally equivalent. The adequacy problem of deciding which equivalence notion to apply is moderated by the fact that the implementation satisfies the B-condition, under which all better known equivalence notions coincide. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Timed Petri nets; Latency time; Place latency nets; Implementation; Pulse generator; Equivalence

Contents

E-mail address: durchholz@gmd.de (R. Durchholz).

0304-3975/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(01)00327-9

1. Introduction

Time is widely held an indispensable addition to Petri nets to make them expressive enough for modelling real world applications like process control, communication protocols, work flow analysis, flexible manufacturing or synchronous circuit design ([2, 7, 9, 13, 16]). Several time extensions have been suggested, which fall broadly into three categories: latency time for tokens on places, duration time for transition firings, and time stamps on tokens. Typical examples are in [13, 7, 5], respectively. The latter is substantially different from the two other ones because it remains within ordinary Petri net theory. In contrast, place and transition time extensions tamper with the firing rule in a serious way because they give up much of event independence. Are they still reducible to elementary Petri nets? Since independence of events is the fundamental motivation for Petri nets, it must be suspected that expressive power is sought at the expense of abandoning the basis of Petri net theory. Coolahan and Roussopoulos warn: "Also, the extensions to Petri nets employed to enhance the general modelling power cause the analytical power to suffer" [2].

The paper proves that such doubts are unjustified, at least for a particular class of latency time extended Petri nets, called place latency systems, and a particular chosen equivalence notion. Neither is the expressive power enlarged, nor is elementary Petri net theory abandoned. The proof is constructive. For each given place latency system, an elementary Petri net system is explicitly specified that has the same behaviour with respect to the chosen equivalence notion.

Knowledge of the basics of Petri net theory is presupposed. Section 2 summarizes what is used and the notation employed.

Section 3 introduces place latency systems as a prototype for the category of Petri nets that are extended by latency times for tokens on places. It also provides a firing rule that formally defines the role of the timing mechanism. Section 4 deals with implementing a place latency system by an elementary net system. It outlines in an informal way the basic ideas to give an intuitive background for the formal definitions to follow in Section 5.

Section 6 is a technical preparation for the proof of the main result. Section 7 presents the main result of the paper, the proof of string equivalence between a place latency system and its implementation. It also reconsiders the significance of the proven equivalence in view of the fact that the implementation satisfies the so-called B-condition, which is a strong indication for the generality of the result. Section 8 concludes the paper with a summary of the results and an outlook to further work.

The present paper is an elaboration of ideas presented in [4].

2. Prerequisites

Mathematics in general

$\mathbb{N}$ is set of natural numbers including 0.

Occasionally, a function f with domain X is canonically extended to the powerset. That is,

$$f(Y) := \{f(x) \mid x \in Y\} \quad \text{for } Y \subseteq X.$$

Such an extension is implicitly understood without extra mention.

"$\ominus$" denotes the symmetric difference, i.e., $a \ominus b = (a - b) \cup (b - a)$.

X^* is the set of all finite sequences over X, including the empty sequence λ. For a sequence $w \in X^*$, *length*(w) is the number of element occurrences in w. An element from X is at the same time considered a sequence of length 1 in X^*.

For a sequence $w \in X^*$, $\ll w \gg$ is the function that maps each element of X to the number of its occurrences in w (*Parikh-function*). Note that $\ll w_1 \gg \; = \; \ll w_2 \gg$ means that w_1 and w_2 are rearrangements of each other.

Frequently, references to formulas are made, together with a given value assignment to the bound variables in the formula. For conciseness and readability, the value bindings of variables are given in a schematic notation. For example, Lemma 26 [w_{fin} for w; c_{eh} for c] is a reference to Lemma 26 with w_{fin} substituted for w, and c_{eh} substituted for c. If a substituted expression happens to be homonymous to the bound variable, this pair is omitted from the binding list. E.g., we write Lemma 26 [w_{eh} for w] rather then Lemma 26 [w_{eh} for $w; c$ for c].

Petri nets

We presuppose the standard basic concepts and terminology of Petri net theory. All nets considered are assumed to be finite, pure and simple, and with no isolated elements. If (P, T, F) is such a net, (P, T, F, c°) is an elementary net system with initial marking c° [11]. A marking is considered a subset of P or the characteristic function of the subset, depending on what is more convenient in a particular context. The values of a characteristic function are chosen to be "$\bullet$" for "marked" and "$\circ$" for "unmarked".

A few concepts and notational conventions that are not so generally used, or are introduced for usage in this paper only, are given below.

The *vicinity* of a net element x is $vic(x) := x^\bullet \cup {}^\bullet x$. The vicinity is also defined for entire transition sequences as $vic(w) := \bigcup\{vic(t) \mid \; \ll w \gg (t) > 0\}$ for $w \in T^*$. Two transitions with disjoint vicinities are said to be *independent*: t_1 *ind* t_2 $:\Leftrightarrow$ $vic(t_1) \cap vic(t_2) = \emptyset$.

Let $m \subseteq P$. The expression $m \triangleright t$ stands for saying that t is enabled in m, as well as for a term denoting the follower marking of m after firing t.[1] Note that $m \triangleright t$ can be used as a term only if $m \triangleright t$ is established as a statement. If we use $m \triangleright t$ in a term within a statement, we implicitly understand the statement $m \triangleright t$ to hold. E.g., $m_1 \triangleright t = m_2$ stands for $m_1 \triangleright t \wedge m_1 \triangleright t = m_2$.

The concepts of enabling independence and follower marking are extended in an obvious way to T^*. For sequences, an associative rule $m \triangleright w_1 w_2 = (m \triangleright w_1) \triangleright w_2$ applies,

[1] We prefer the infix notation to the usual bracketed one for better readability of nested expressions.

which is often used tacitly. Further we define: σ is a *firing sequence in m* iff $m \triangleright \sigma$. σ is a *firing sequence* iff σ is a firing sequence in some marking m. σ is an *initial firing sequence* iff σ is a firing sequence in c°.

A *case* is a marking $c^\circ \triangleright \sigma$, where σ is some initial firing sequence. Thus, c° is itself a case. Occasionally, we say *initial case* for the initial marking.

We note two simple properties of elementary nets, with tags for ease of reference:

(Net1) $\quad \forall w_1, w_2 \in T^*, \quad m \subseteq P(m \triangleright w_1 \wedge m \triangleright w_2 \wedge w_1 \; ind \; w_2 \Rightarrow m \triangleright w_1 w_2)$.

If two independent sequences are both enabled at a marking, so is their concatenation. (In fact any interleaving, but we need this weak form only.)

(Net2) $\quad \forall t \in T, \quad m, d \subseteq P(m \triangleright t \Rightarrow \|(m \triangleright t) \cap d\| = \|m \cap d\| - \|{}^\bullet t \cap d\| + \|t^\bullet \cap d\|)$.

On firing an enabled transition, the cardinality of the marking is decreased by the cardinality of the pre-set, and increased by the cardinality of the post-set of the transition. This holds also for any projection onto a subset d of P.

3. Place latency nets

A place latency net is basically an ordinary net with “latencies” attached to its places. In addition, a “pulse generator” π is assumed that has an impact on the firing of transitions. The intention is that a token arriving at a place is unavailable for as many pulse ticks as the latency says. After that, it is treated in the usual way. Further, two kinds of transitions, “regular” and “hot” ones, are distinguished, which behave differently, as specified by the firing rule (via the enabling rule). It puts hot transitions, and only these, in relation to pulses by requiring that any enabled hot transition has priority over the pulse generator.

Place latency nets and associated concepts are now formally defined.

(P, T, F, H, l) is a *place latency net* iff (P, T, F) is an ordinary net, $H \subseteq T$, and $l : P \to \mathbb{N}$. Elements of H are called *hot transitions*, l is called the *latency function* of the place latency net. A transition from $T - H$ is called *regular*. Elements of P are called *latency places* to emphasize the context.

$(P, T, F, H, l, c^\circ_{\mathrm{lat}}, \pi, \circ)$ is a *place latency system* iff (P, T, F, H, l) is a place latency net, $\pi \notin T$, $\circ \notin \mathbb{N}$, and $c^\circ_{\mathrm{lat}} : P \to \mathbb{N} \cup \{\circ\}$ such that $c^\circ_{\mathrm{lat}}(s) = \circ$ or $c^\circ_{\mathrm{lat}}(s) \leqslant l(s)$ for all $s \in P$. c°_{lat} is called the *initial marking*, and π the *pulse generator* of the place latency system. Generally, any function $m : P \to \mathbb{N} \cup \{\circ\}$ such that $m(s) = \circ$ or $m(s) \leqslant l(s)$ for all $s \in P$ is a *marking* of (P, T, F, H, l). By definition, π is *not* a transition. However, to avoid awkward phrasing, we say somewhat sloppily “transition” also when we mean an element of $T \cup \{\pi\}$. To emphasize that a transition is not π, it is occasionally called a *proper transition*.

If for $s \in P$ a marking m yields a natural number $(m(s) \neq \circ)$, we say that s ***carries a token in*** m. The token is said to be *available* or *unavailable* if $m(s) = 0$ or $m(s) > 0$, respectively. To visually emphasize available tokens, and also to correspond to usual Petri net graphic symbolism, we mostly write $m(s) = \bullet$ for $m(s) = 0$. Generally, we write “$\bullet$” instead of “0” in its role as a value of a marking. Much as for elementary nets,

$m(s) = \bullet$ means s is marked by an available token, and $m(s) = \circ$ means s is unmarked. Place latency nets have, however, additional marking values that are neither $\bullet$ nor $\circ$, but positive integers for "marked by an unavailable token". Note that $m(s) > 0$ is equivalent to $m(s) \notin \{\circ, \bullet\}$. A marking value $m(s) \geqslant 0$ of a latency place s is called the (token) *unavailability* of s (at m). The value "0" is included, although this somewhat stretches the term.

The enabling rule for proper transitions is much as for ordinary Petri nets. In contrast, its extension to π introduces a global aspect through its use of a quantifier. We define

$$m \rhd t \quad \text{iff } m({}^\bullet t) = \{\bullet\} \quad \text{and} \quad m(t^\bullet) = \{\circ\} \quad \text{for } t \in T,$$

$$m \rhd \pi \quad \text{iff } \neg \exists\, h \in H(m \rhd h).$$

If $m \rhd x$ for $x \in T \cup \{\pi\}$, then x is said to be *enabled* in m. Note that an unavailable token obstructs enabling in the pre-set as well in the post-set of a proper transition. π is not enabled as long as there are enabled hot transitions.

A *firing rule* is now given, which defines follower markings.

$$m_1 \rhd t = m_2 \quad \text{iff } m_1 \rhd t \text{ and } \forall s \in P$$

$$\left(m_2(s) = \begin{cases} \circ & \text{if } t \in s^\bullet \\ l(s) & \text{if } t \in {}^\bullet s \\ m_1(s) - 1 & \text{if } t = \pi, m_1(s) > 0 \\ m_1(s) & \text{otherwise} \end{cases} \right).$$

If $m \rhd t$ holds for $t \in T \cup \{\pi\}$, the marking $m \rhd t$ is said to be the *follower marking* of m under t. Just as for Petri nets, the expression $m \rhd t$ is used for the statement of enabledness, as well as for the term denoting the follower marking.

The firing rule says that, on firing an enabled $t \in T$, the places in the pre-set of t become unmarked, the places in the post-set of t become marked with maximal unavailability. All others retain their markings unchanged. On firing π, all positive unavailabilities are decremented, while unmarked and availably marked places retain their markings.

Note that firing of π does not require any proper transition to be enabled. Note also that a hot transition does not have priority over regular transitions. The firing rule itself does not distinguish between enabled hot and regular transitions. As a consequence, a regular transition may fire when it is in conflict with, hence disables, a hot transition, which in turn may cause π to become enabled. Thus, an enabled hot transition may become disabled rather than fire before π. More generally, proper transitions may repeatedly become disabled and enabled without π firing. If this happens to the only enabled hot transition, π may become enabled and disabled any number of times without firing itself. Conversely, π may never become enabled, e.g., if the system loops with hot transitions and zero latency.

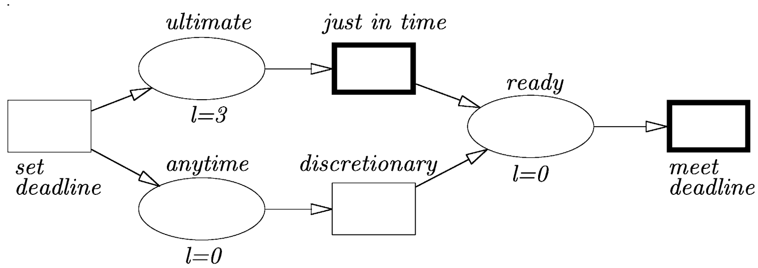


Fig. 1. Example of a place latency net.

The global aspect introduced by the π enabling rule is carried over to the firing rule because the latter makes reference to enabling. In addition, for $t=\pi$ in the firing rule, the check for positive unavailability on the markings of all latency places introduces another global aspect. (For $t\in T$, the firing rule is local because the quantifier is restricted to the vicinity of t, if a change is at all required.) Globality is the critical feature that makes place latency nets special. The pulse generator forces a strong synchronization upon the net.

The concepts of enabling and follower marking are extended to sequences $w\in(T\cup\{\pi\})^*$ in the obvious way. The system-related notions *initial sequence* and *case* are defined accordingly.

In diagrams, we depict latency places by ovals with associated latency, like $\bigcirc^{l=3}$. A rectangle with a normal or heavy outline depicts a regular or hot transition, respectively: $\square$ for a regular transition, $\boxed{}$ for a hot transition.

In Fig. 1, an illustrative example is given for a place latency system with

$P=\{\mathit{ultimate},\ \mathit{anytime},\ \mathit{ready}\}$,

$T=\{\mathit{set\ deadline},\ \mathit{just\ in\ time},\ \mathit{discretionary},\ \mathit{meet\ deadline}\}$,

F as expressed by the arrows,

$H=\{\mathit{just\ in\ time},\ \mathit{meet\ deadline}\}, l(\mathit{ultimate})=3,\ \ l(\mathit{anytime})=l(\mathit{ready})=0$,

$c^0_{\mathrm{lat}}(P)=\{\circ\}$.

It describes a place latency system for deadlined action. The upper branch takes care that the deadline is not missed. Three pulses after firing *set deadline* the token is available in *ultimate*. If enabled, the hot transition *just in time* then fires before the next pulse. Since $l(\mathit{ready})=0$, the token in *ready* is immediately available, so that the hot transition *meet deadline* must also fire before the next pulse. Thus, *meet deadline* occurs before the fourth pulse. The lower branch allows for earlier action much as in an ordinary Petri net. Note that, even if *discretionary* is first, *just in time* still occurs before the fourth pulse tick because it is enabled after *meet deadline* has fired. No provision is made for an orderly disposal of the token left behind since this is of no particular concern for the demonstration of how a place latency system performs.

One could as well consider *just in time* and *discretionary* together as representing the event of meeting the deadline. The summarizing *meet deadline* is not really needed as an explicit transition. But it provides a good opportunity for demonstrating that hot transitions enabled in sequence are not separated by π if only zero latency is involved.

It is worth mentioning that the firing rule just specifies the possible order of transition firings. It is a matter of view whether one considers π to be kept from firing by enabled hot transitions, or an enabled hot transition to be forced to fire prior to π. In the example above, the interpretation puts emphasis on forced firing. Such an interpretation turns generally up when time-outs are modelled, as, e.g., in communication protocols to decide that a message has to be considered lost [1]. In the forced firing interpretation, the pulse generator is seen as ticking invariably ahead. A device modelled by the place latency system obeys the forced firing specification if its functional units modelled by hot transitions are early enough relative to the functional unit described by π. The difference between the views and the role and understanding of violations of a deadline requirement are discussed in some detail in [3].

When we model place latency systems, forcing must also be expressed in the implementation. With respect to the possibility of representing transition forcing, [5] states: "In a Petri net, the only way to (say) force a set of transition to fire is to disable the other ones". This is what we will do when we implement place latency systems (Sections 4 and 5). However, the idea of indirect forcing is not due to the use of Petri nets. As we have noted above, it does not pertain to the formal system, but to its interpretation. Most interestingly, for place latency systems it comes into play at the very point where Petri nets are abandoned. The quantified extension of the firing rule to include π is alien to net theory, but it is exactly this which provides the formal texture for the forced firing interpretation. In the interpretation, it must of course be expressed with net-theoretical means. This is achieved in the sense of the above quotation from Ghezzi et al. [5] by a mechanism which disables the transition that represents the pulse generator while a hot transition is enabled.

Apart from the more obvious characteristics of place latency nets, there is also a subtle aspect involved. The pulse generator can be seen as a device that classifies certain transition occurrences into pre-pulse and post-pulse events. Intersections of classes from different pulses can be interpreted as constituting time intervals [10]. Within an interval between subsequent pulses, transition occurrences are still partially ordered as usual, but there is no time metrics (this reminds somewhat of the "micro time scale" in VHDL circuit simulation [12]).

The decision of whether a given transition occurrence falls into some pre- or post-class is made in a strangely ambiguous context. As noted above, π may become enabled and disabled any number of times without firing itself. This may occur also while a conflict or its resolution between a regular transition and π is immanent. Thus, when the transition actually fires, it is uncertain, whether this happens in an unquestionable winning position, or whether an arbitration was needed.

This allusion of an intriguing situation can be given a precise formal characterization. We will come back to this towards the end of the next section, when we are able to discuss it in a net-theoretic setting for the elementary net system Σ_{impl} that models a given place latency systems (see Section 5).

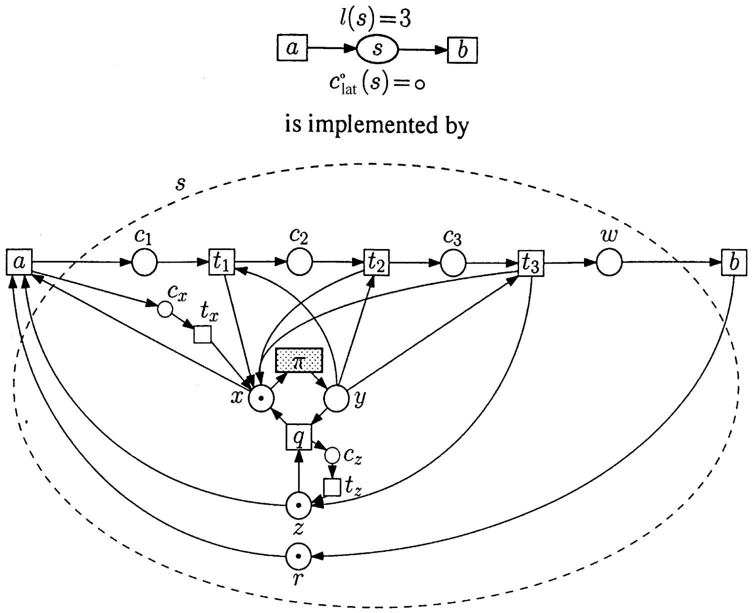


Fig. 2. Expansion of a place with latency 3.

4. Introduction to place latency system implementation

In this section, an informal description is given of how to generate an elementary net system for a given place latency system. It first considers the expansion of a single latency place. Subsequently it deals with the restrictions imposed by a single hot transition. Finally, an example of the implementation of a complete place latency system is given. Expansions of latency places and hot transition restrictions are basically local, except for the pulse generator π as the only element with a global impact.

A latency place is expanded into a place-bounded subnet. The diagrams in Fig. 2 show a miniature place latency system and its implementation. The shaded transition π does not belong to the expansion of the place latency system because in larger nets π is shared by the expansions of all places with positive latencies. The dashed oval outlines the sub-net generated by the expansion of s. It is not part of the implementation.

Note that we have used the pulse generator π from the place latency net as an abstract object for constructing the implementation. This is an arbitrary, but convenient, choice since it expresses an intended correspondence. Hopefully, no confusion arises about the different roles of π in the original place latency net as an add-on concept, and its implementation, where it is an ordinary transition. Similarly, a and b are directly used in the implementation. Place s, however, does not reappear in the implementation.

By executing the diagram one can convince oneself that the implementation behaves like the original place latency system if we consider auxiliary transitions to be invisible (τ-transitions). In terms of regular expressions, the visible behaviour is $(\pi^* a \pi^3 \pi^* b)^*$.

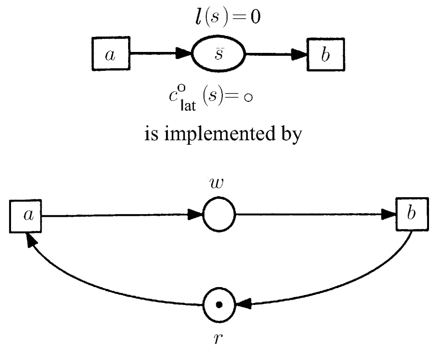


Fig. 3. Expansion of a zero latency place.

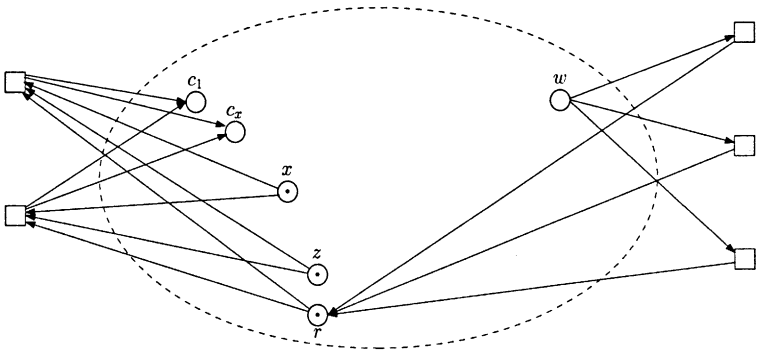


Fig. 4. Latency place expansion in a larger vicinity.

In particular, the following holds for both, the transitions together with π of the place latency net, and the visible transitions of its implementation.

- Initially, only a and π may fire until a has occurred.
- After occurrence of a and three subsequent occurrences of π, b may fire, unaffected by any further occurrences of π.
- a may next fire only after an occurrence of b.
- Occurrence of π is not restricted while a or b are enabled.

In the above example a place with latency 3 has been considered. The generalization to an arbitrary latency is obvious. It is obtained by just fitting the length of the "delay section" $(c_1 \rightarrow t_1 \rightarrow c_2 \rightarrow t_2 \rightarrow c_3 \rightarrow t_3)$ to $l(s)$. This works even for latency 1. For zero latency the structure collapses. Its implementation is given in Fig. 3.

So far, the latency places in the examples have one input and one output transition. In general, there can be any number of transitions in the vicinity of a latency place. This is handled by making multiple connections to the entry and exit places of the latency place expansion. Fig. 4 exemplifies the embedding of a latency place expansion into a larger vicinity. Only the boundary places are shown.

Table 1
Unavailability represented by expansion markings

Marking of latency place	Representation in expansion
○	$\{x, z, r\}$, $\{y, z, r\}$, $\{x, c_z, r\}$, $\{y, c_z, r\}$,
3	$\{c_x, c_1\}$, $\{x, c_1\}$
2	$\{y, c_1\}$, $\{x, c_2\}$
1	$\{y, c_2\}$, $\{x, c_3\}$
•	$\{x, z, w\}$, $\{y, z, w\}$, $\{x, c_z, w\}$, $\{y, c_z, w\}$, $\{y, c_3\}$

In Fig. 2, the marking of the expansion is not the only one that represents the unmarked latency place. E.g., $\{x, c_z, r\}$ would serve as well. Although a is then not directly enabled, it becomes so on firing t_z, which is considered invisible. Thus, an observer could not decide whether a does not fire on its own, or because it has to wait for t_z. In general, any marking value of a latency place has several representations in the expansion. Table 1 shows all possible representations for the example.

Though there is a direct correspondence between the unavailability of a token on the latency place and the places c_1, c_2, c_3 and w of the expansion, these places alone are not sufficient for a correct distinction of the represented markings. This is because the switch from some unavailability value to the next smaller one must occur coincidentally with π, and only then. From this follows that, e.g., c_1 together with y represents the same unavailability as does c_2 together with x, because no π occurrence goes in between. Similarly, token availability cannot be determined from the marking on w alone. $\{y, c_3\}$ anticipates the availability since it takes "no time" (no occurrence of π) for t_3 to occur.

Though the last transition in the "delay section" of the latency place expansion, t_3 in the example, does not contribute to availability, it plays an important technical role in the system implementation since its firing is the final event that readies the expansion for external use. We call it the *exit transition* of the latency place expansion. Expansions of places with zero latency do not have exit transitions.

If the system cycles in a group of markings that contain r (○-row) or a group of markings that contain w (•-row except $\{y, c_3\}$), this means that pulses go by unused. For positive unavailability, an unused pulse is not admissible because of the requirement for a latent token to "age" (decrease unavailability) on each occurrence of π. In the implementation, this is reflected by the fact that token migration on the delay section is interlocked with occurrences of π.

Next we consider hot transitions. The enabling rule for π requires that no hot transition is enabled. Thus, the construction must take care that the representation of the pulse generator is disabled when a hot transition is enabled. More precisely, an inhibition of π must be implemented which becomes effective coincidentally with the first hot transition becoming enabled, and becomes ineffective coincidentally with the last hot transition becoming disabled.

For each hot transition h, the inhibition is achieved by a sub-net, called the *hot expansion* of h, that maintains a "deficiency count" for h. The term "hot expansion of h" is a bit sloppy since, literally speaking, h itself is not expanded, but rather reappears in the implementation. It means, more precisely, "expansion of the restriction effected by h". The *deficiency count* (short dc) tells "how far" the transition is from being enabled, that is, how many places in its pre-set do not carry an available token, and how many places in its post-set carry a token. (Note the asymmetry for pre- and post-set in the requirement for token availability.) π should be inhibited by the hot expansion for $h \in H$ if and only if its deficiency count is zero.

If π is inhibited, it must not be enabled. The converse is not true. If π is disabled, it is not necessarily inhibited, because it may become enabled through firing of τ-transitions. Moreover, π may remain uninhibited while gaining and loosing enabledness several times by firings of τ-transitions. In this context it may be appropriate to point out that this paper deals with *possible* firing sequences rather than with probabilities. (In fact, under the unwarranted view of a (say) uniform distribution of probabilities for transition firings, enabling of π would become extremely unlikely.)

Fig. 5 shows the inhibition mechanism for a hot transition h with three latency places s_1, s_2, s_3, in its vicinity. To reduce the graphical complexity of place latency expansions, t_{xs_1}, t_{xz_1}, t_{xs_2} and t_{zs_2} are represented implicitly only as parts of the double-headed arrows denoting pseudo side-conditions. In the given marking, h is enabled and the deficiency count is zero. Also, for the sake of a clearer visualization, the shaded rectangle for π is drawn in several copies to avoid convoluted arcs. Please keep in mind that they all mean the same transition.

By executing the example step by step, the reader may find that, in fact, the construction enforces priority of the enabled hot transition over π, and that otherwise it imposes no restrictions on the sequence of other transitions, taking into account that the tokens within the hot expansion may "float freely" by firings of invisible transitions. One may also find that the number of tokens on the hot expansion, i.e., on $\{c_{fh}, f_h, dc_{s_1h}, dc_{s_2h}, dc_{s_3h}\}$ equals the deficiency count. This is generally true and is referred to as the "deficiency count property", which is proven in Section 6.5.

The present section concludes with an example that integrates the construction elements discussed above. It is the implementation of the deadline example from Section 3.

Even with this simple example, the implementation illustrates two typical aspects of place latency systems, complexity and global synchronization.

Global synchronization is expressed by the ubiquity of the shaded rectangles. Remember it denotes the same transition π wherever it occurs. One could appropriately imagine only one shaded rectangle sitting above the entire diagram, with arcs connecting it to each positive latency place expansion, and to each hot expansion.

Though high complexity is a well-known facet for modelling with elementary Petri nets, its actual degree for even such a simple place latency system might be surprising. One may suspect that this is due to an unskilled implementation, but we believe that the given construction can hardly be simplified (except for one transition

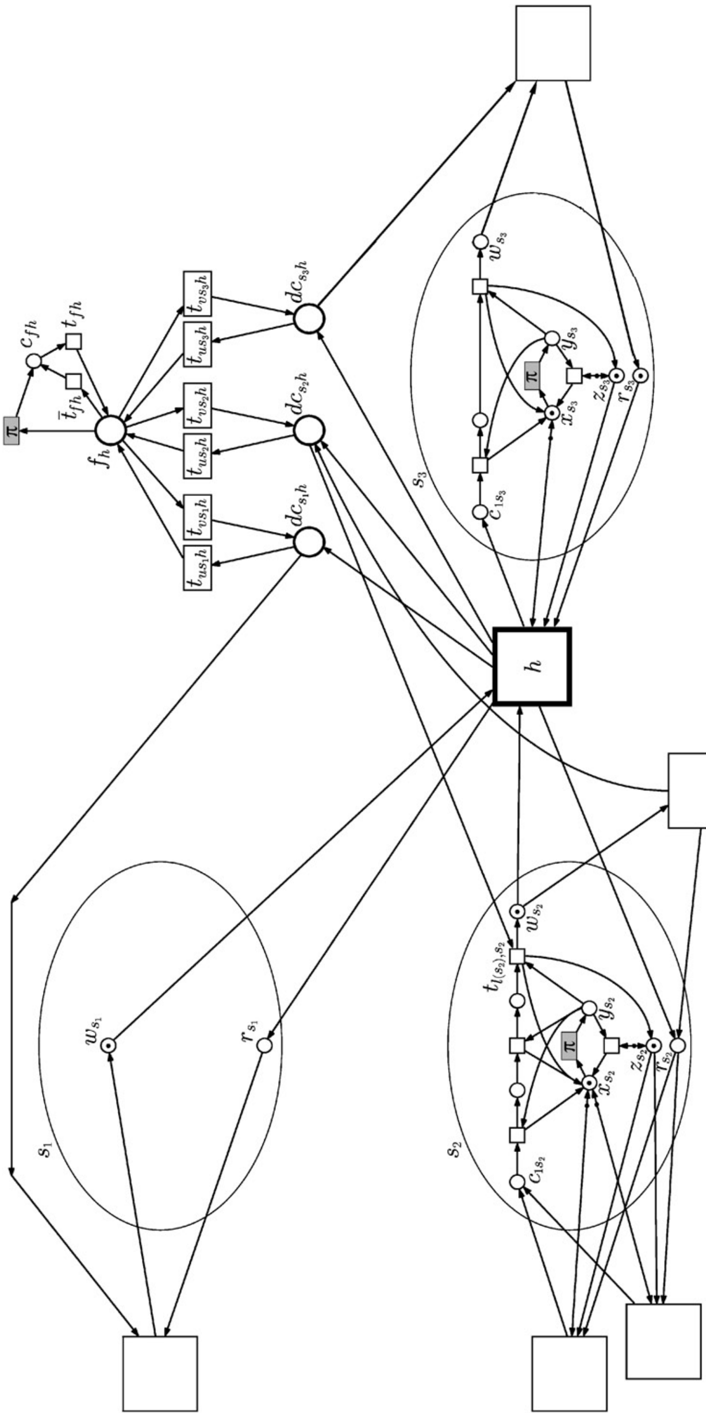


Fig. 5. π restriction by enabled hot transition.

in the hot expansion which is included to make some proofs more homogeneous). Given that such complexity is intrinsic to latency representation, the implementation brings it into the open. One can look at it as an illustration of a note in [5] with respect to forced firing: "Therefore, the firing condition cannot be determined by locally examining the transition, but it would require a view of the whole net, ..."

However, the complexity also shows that for practical purposes concepts for concisely describing the impact of a clock, as provided by the various timed net approaches, are indispensable for making manageable descriptions of time-driven systems where this kind of synchronization is intended. Thus, time extensions can be seen as powerful descriptive means for handling a particular kind of complex synchronization specifications. Their translation to elementary net systems, as done in this paper, is of purely theoretical interest. No direct practical application is intended with representing place latency systems by elementary nets, but the theoretical acceptability is significant for the practical use of time-controlled approaches. It is the very mixture of concurrency and strictly clocked control that makes Petri nets with time extensions interesting for some applications, in particular for complex circuit description. Shadad et al. [12] puts it like this: "Time is one of the most important aspects of a hardware description language because the timing characteristics of hardware are perhaps the most difficult to represent in text. In part, this difficulty is due to the massive parallelism that may exist in a hardware description".

With an overall outline of how implementations of place latency nets look like, we are now able to discuss in net-theoretical terms the ambiguities of event ordering that were mentioned in Section 3. Generally, such ambiguities occur in situations in which an independent transition interferes with appearance or resolution of conflicts. This is called a *confusion*. Typically, arbiters for deciding the winner of a race may run into confused situations. In [14], arbiters have been proven to inherently contain the possibility of confusion. Though, by appropriate design, a confusion can be confined to an ever smaller context, it can never entirely be dissolved. Formal definitions of confusion can be found in the literature (e.g. [11] or [14]).

In our case, a race takes place between a transition to occur within the current interval, and the pulse generator to begin a new one. In the implementation examples above, the pulse generator is involved in confusions during decisions about the pulse interval in which proper transitions fire. There are mainly two such situations, (i) determining the pulse interval in which a newly placed token starts ageing, and (ii) determining the pulse interval in which a hot transition fires. In terms of a notation suggested in [11], a confusion example for (i) is $(c, \textit{set deadline}, t)$ in Fig. 6, where c is a marking such that *set deadline* is enabled, but π is disabled, and t is a transition from a hot expansion that is enabled in c and enables π when it fires. Note that t is independent of *set deadline* by construction. In this implementation, the confusion is confined to what seems a marginal pattern. The important fact is that it is still there. This illustrates the remark above about persistence of confusions in attempts to confine them to smaller contexts.

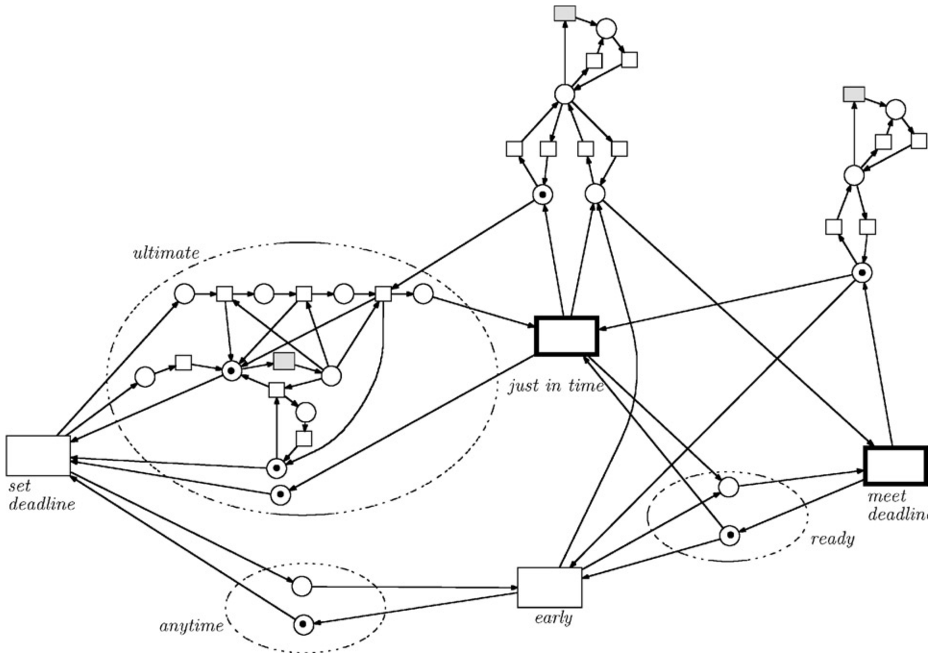


Fig. 6. Implementation of the deadline place latency net.

5. Formal place latency system implementation

This section describes how from a given place latency system $\Sigma_{\rm lat} = (P_{\rm lat}, T_{\rm lat}, F_{\rm lat}, H, l, c^{\circ}_{\rm lat}, \pi, \circ)$ an elementary net system $\Sigma_{\rm impl} = (P_{\rm impl}, T_{\rm impl}, F_{\rm impl}, c^{\circ}_{\rm impl})$ can be derived as an implementation. The derivation is constructive, so that, in principle, it is possible to generate it mechanically. However, this would be of little practical interest, as has already been mentioned in the previous section. In the next section, we show that $\Sigma_{\rm impl}$ is equivalent to $\Sigma_{\rm lat}$ in a quite broad sense.

To create the implementation, many new abstract objects are introduced. We understand that in the following definitions different symbols mean different objects, and that they are also different from any other objects introduced so far. This applies also to indexed symbols. In particular, there are symbols that are indexed by elements from $\Sigma_{\rm lat}$. They mean different objects if the indexes are different. E.g., $c_{xs_1} \neq c_{xs_2}$ is understood for $s_1 \neq s_2$.

To provide building blocks for the implementation, we pre-define generic structures that are independent of the particular given $\Sigma_{\rm lat}$. The generic structures can easily be recognized in the examples of the previous section.

$$P^{\rm G}_0 := \{w, r\}, \quad T^{\rm G}_0 := \{\}, \quad F^{\rm G}_0 := \{\}.$$

For $i>0$:

$$
\begin{aligned}
P_i^{\mathrm{G}} &:= \{w,x,y,z,r,c_x,c_z\} \cup \{c_v \mid 1 \leqslant v \leqslant i\},\\
T_i^{\mathrm{G}} &:= \{q,t_x,t_z\} \cup \{t_v \mid 1 \leqslant v \leqslant i\},\\
F_i^{\mathrm{G}} &:= \{(y,q),(q,x),(z,q),(q,c_z),(c_z,t_z),(t_z,z),(t_i,w),(t_i,z),(c_x,t_x),(t_x,x)\} \cup\\
&\qquad \{(c_v,t_v) \mid 1 \leqslant v \leqslant i\} \cup\\
&\qquad \{(t_v,c_{v+1}) \mid 1 \leqslant v \leqslant i-1\} \cup\\
&\qquad \{(y,t_v) \mid 1 \leqslant v \leqslant i\} \cup\\
&\qquad \{(t_v,x) \mid 1 \leqslant v \leqslant i\}.
\end{aligned}
$$

The generic structures are now used for defining specific objects for a given Σ_{lat}.

Latency place expansion

It will frequently be necessary to distinguish between places with zero and positive latency. For this, we define $P_{\mathrm{lat}}^0 := \{s \in P_{\mathrm{lat}} \mid l(s)=0\}$, $P_{\mathrm{lat}}^+ := \{s \in P_{\mathrm{lat}} \mid l(s)>0\}$.

For $s \in P_{\mathrm{lat}}$ we define:

$$P_s := \{p_s \mid p \in P_{l(s)}^{\mathrm{G}}\}, \quad T_s := \{t_s \mid t \in T_{l(s)}^{\mathrm{G}}\},$$

$$F_s := \begin{cases} \emptyset & \text{if } s \in P_{\mathrm{lat}}^0,\\ \{(\xi_s,\eta_s) \mid (\xi,\eta) \in F_{l(s)}^{\mathrm{G}}\} \cup \{(x_s,\pi),(\pi,y_s)\} & \text{if } s \in P_{\mathrm{lat}}^+. \end{cases}$$

A latency place expansion is a fairly local sub-net of the net to be constructed. As mentioned already in Section 4, the sole global impact is focussed in its connection to the pulse generator π. π itself is not considered to belong to the latency place expansion.

Hot expansion

Each hot transition h of Σ_{lat} gives rise to a sub-net called "hot expansion" that interconnects π and h with the expansions of all places in the vicinity of h in Σ_{lat}. Thus, except for π, a hot expansion is also local, though it is not as narrowly restricted as a latency place expansion.
For $h \in H$:

$$P_h := \{dc_{sh} \mid s \in vic(h)\} \cup \{f_h, c_{fh}\}$$

$$T_h := \{t_{ush} \mid s \in vic(h)\} \cup \{t_{vsh} \mid s \in vic(h)\} \cup \{t_{fh}, \bar{t}_{fh}\}$$

$$
\begin{array}{lll}
F_h := & \left.\begin{array}{l}\{(dc_{sh}, t_{ush}) \mid s \in vic(h)\} \cup \\ \{(t_{ush}, f_h) \mid s \in vic(h)\} \cup \\ \{(f_h, t_{vsh}) \mid s \in vic(h)\} \cup \\ \{(t_{vsh}, dc_{sh}) \mid s \in vic(h)\} \cup \\ \{(c_{fh}, t_{fh}), (t_{fh}, f_h)\} \cup \\ \{(f_h, \bar{t}_{fh}), (\bar{t}_{fh}, c_{fh}),\} \cup \end{array}\right\} & \begin{array}{l}\text{tokens may float freely in } P_h \\ \text{without changing } dc\end{array} \\
& \{(f_h, \pi), (\pi, c_{fh})\} \cup & \pi \text{ needs transiently a token from } P_h \\
& \{(dc_{sh}, t_{l(s),s}) \mid s \in {}^{\bullet}h \wedge s \in P^+_{\mathrm{lat}}\} \cup & \text{decreasing } dc \text{ when tokens become available on positive latency input places} \\
& \{(dc_{sh}, t) \mid s \in {}^{\bullet}h \cap t^{\bullet} \wedge s \in P^0_{\mathrm{lat}}\} \cup & \text{decreasing } dc \text{ on marking zero latency input places} \\
& \{(dc_{sh}, t) \mid s \in h^{\bullet} \cap {}^{\bullet}t\} \cup & \text{decreasing } dc \text{ on unmarking output places} \\
& \{(t, dc_{sh}) \mid s \in {}^{\bullet}h \cap {}^{\bullet}t \wedge t \neq h\} \cup & \text{increasing } dc \text{ on unmarking input places} \\
& \{(t, dc_{sh}) \mid s \in h^{\bullet} \cap t^{\bullet} \wedge t \neq h\} \cup & \text{increasing } dc \text{ on marking output places} \\
& \{(h, dc_{sh}) \mid s \in vic(h)\} & \text{setting } dc \text{ to maximum value on firing } h.
\end{array}
$$

System implementation

To finalize system implementation, the sub-nets defined so far are composed into a single net for which an initial marking is defined.

The composition is prepared by identifying for each latency place the expansion places that link the expansion to its environment in four different ways:

$$
\mathit{InPre}_s := \begin{cases} \{r_s\} & \text{if } s \in P^0_{\mathrm{lat}} \\ \{x_s, z_s, r_s\} & \text{if } s \in P^+_{\mathrm{lat}} \end{cases} \quad \text{input places for } {}^{\bullet}s
$$

$$
\mathit{OutPre}_s := \begin{cases} \{w_s\} & \text{if } s \in P^0_{\mathrm{lat}} \\ \{c_{x,s}, c_{1,s}\} & \text{if } s \in P^+_{\mathrm{lat}} \end{cases} \quad \text{output places for } {}^{\bullet}s
$$

$$
\mathit{InPost}_s := \begin{cases} \{w_s\} & \text{if } s \in P^0_{\mathrm{lat}} \\ \{w_s\} & \text{if } s \in P^+_{\mathrm{lat}} \end{cases} \quad \text{input places for } s^{\bullet}
$$

$$
\mathit{OutPost}_s := \begin{cases} \{r_s\} & \text{if } s \in P^0_{\mathrm{lat}} \\ \{r_s\} & \text{if } s \in P^+_{\mathrm{lat}} \end{cases} \quad \text{output places for } s^{\bullet}.
$$

With the help of these auxiliary sets we are now ready to define the net of Σ_{impl}.

$$P_{\text{impl}} := \bigcup\{P_s \mid s \in P_{\text{lat}}\} \cup \bigcup\{P_h \mid h \in H\}$$

$$T_{\text{impl}} := \bigcup\{T_s \mid s \in P_{\text{lat}}^{+}\} \cup \bigcup\{T_h \mid h \in H\} \cup T_{\text{lat}} \cup \{\pi\}$$

$$F_{\text{impl}} := \bigcup\{F_h \mid h \in H\} \cup \bigcup\{F_s \cup (\mathit{InPre}_s \times {}^{\bullet}s) \cup ({}^{\bullet}s \times \mathit{OutPre}_s) \cup$$
$$(\mathit{InPost}_s \times s^{\bullet}) \cup (s^{\bullet} \times \mathit{OutPost}_s) \mid s \in P_{\text{lat}}\}.$$

To obtain Σ_{impl}, an initial case must still be defined. As can be seen from the examples in Section 4 (Fig. 3 for latency place expansion, Fig. 5 for hot expansion), a marking of Σ_{lat} cannot be expected to be uniquely representable in Σ_{impl}. For the sake of being definite, we select from several candidates an arbitrary "normalized" one for the initial case.

$$c^{\circ}_{\text{impl}} := \{r_s \mid c^{\circ}_{\text{lat}}(s) = \circ\} \cup \{w_s \mid c^{\circ}_{\text{lat}}(s) = \bullet\} \cup \{x_s \mid l(s) > 0\} \cup$$
$$\{z_s \mid l(s) > 0 \wedge c^{\circ}_{\text{lat}}(s) \in \{\circ, \bullet\}\} \cup$$
$$\{c_{is} \mid 1 \leqslant i \leqslant l(s) \wedge c^{\circ}_{\text{lat}}(s) = l(s) + 1 - i\} \cup$$
$$\{dc_{sh} \mid c^{\circ}_{\text{lat}}(s) = \circ \wedge h \in H \cap s^{\bullet}\} \cup$$
$$\{dc_{sh} \mid c^{\circ}_{\text{lat}}(s) = \bullet \wedge h \in H \cap {}^{\bullet}s\} \cup$$
$$\{dc_{sh} \mid c^{\circ}_{\text{lat}}(s) \notin \{\circ, \bullet\} \wedge h \in H \cap \mathit{vic}(s)\}.$$

For the initial case, token presence and availability for a latency place s can easily be determined just from looking at places r_s and w_s of its expansion. The following lemma follows immediately from the definition of c°_{impl}. It also contains a statement about mutual exclusion of r_s and w_s.

Lemma 1 (Criteria for initial case representation).
$\forall s \in P_{\text{lat}}((r_s \in c^{\circ}_{\text{impl}} \Leftrightarrow c^{\circ}_{\text{lat}}(s) = \circ) \wedge (w_s \in c^{\circ}_{\text{impl}} \Leftrightarrow c^{\circ}_{\text{lat}}(s) = \bullet) \wedge (r_s \notin c^{\circ}_{\text{impl}} \vee w_s \notin c^{\circ}_{\text{impl}}))$.

It will turn out later (Lemma 5) that $c_{\text{lat}}(s) = \bullet$ does not imply $w_s \in c_{\text{impl}}$ for an arbitrary case $c_{\text{lat}} \subseteq P_{\text{lat}}$ and its representation c_{impl}. For the initial case, however, the "normalization" adopted for its representation makes $c^{\circ}_{\text{lat}}(s) = \bullet \Rightarrow w_s \in c^{\circ}_{\text{impl}}$ true.

6. Features and properties of the implementation

Before we set out to establish formal relations between Σ_{lat} and its implementation Σ_{impl}, we explore in this section properties of Σ_{impl} by itself. This does, of course, not mean that Σ_{lat} is entirely disregarded. As a defining part, it naturally appears in almost all considerations of Σ_{impl}. Therefore, actually both systems are considered in parallel, so that we have to be careful to make always clear to which system we refer when we use concepts like "$\triangleright$" that apply to both.

6.1. Basic notions

This section provides a number of concepts and abbreviations which are subsequently employed for dealing with Σ_{impl}.

First, some symbols that denote often used subsets of T_{impl} and P_{impl} are introduced. Note that the defining terms for the first four of them are literal parts of the definitions of T_{impl} and P_{impl}.

$$
\begin{array}{ll}
P_{\mathrm{placeexp}} := \bigcup\{P_s \mid s \in P_{\mathrm{lat}}\} & \text{"latency place expansion places"} \\
T_{\mathrm{placeexp}} := \bigcup\{T_s \mid s \in P^+_{\mathrm{lat}}\} & \text{"latency place expansion transitions"} \\
P_{\mathrm{hotexp}} := \bigcup\{P_h \mid h \in H\} & \text{"hot expansion places"} \\
T_{\mathrm{hotexp}} := \bigcup\{T_h \mid h \in H\} & \text{"hot expansion transitions"} \\
P_{H'} := \bigcup\{P_h \mid h \in H' \cap H\} & \text{"restricted hot expansion places"} \\
T_{H'} := \bigcup\{T_h \mid h \in H' \cap H\} & \text{"restricted hot expansion transitions"} \\
T_v := T_{\mathrm{lat}} \cup \{\pi\} & \text{"non-}\tau\text{-transitions (visible)"} \\
T_\tau := T_{\mathrm{placeexp}} \cup T_{\mathrm{hotexp}} & \text{"}\tau\text{-transitions (invisible)"} \\
T_{\mathrm{exit}} := \{t_{l(s),s} \mid s \in P^+_{\mathrm{lat}}\} & \text{"latency place expansion exit transitions".}
\end{array}
$$

Note that $T_{\mathrm{impl}} = T_v \cup T_\tau$, and that all unions involved are disjoint. Further, for $P_{H'}$ and $T_{H'}$, we will in practice always have $H' \subseteq H$. We have chosen not to constrain the definition for reasons of formal conciseness only.

For conciseness of expression, we use a symbol for the set of all cases of Σ_{impl}:

$$C_{\mathrm{impl}} := \{c^\circ_{\mathrm{impl}} \triangleright w \mid w \in T^*_{\mathrm{impl}} \wedge c^\circ_{\mathrm{impl}} \triangleright w\} \quad \text{"cases of } \Sigma_{\mathrm{impl}}\text{".}$$

Special notations are introduced for Σ_{impl} for the concepts of pre-set, post-set and vicinity. For Σ_{lat}, the usual notation "$\bullet$" and "*vic*" is retained. For other concepts, distinguishing symbols are not introduced because the system to which they refer is normally clear from the context. Where necessary, Σ_{impl} and Σ_{lat} are explicitly mentioned.

$$
\begin{array}{ll}
{}^{\blacksquare}x := \{y \mid (y,x) \in F_{\mathrm{impl}}\}, & \text{"pre-set in } \Sigma_{\mathrm{impl}}\text{"} \\
x^{\blacksquare} := \{y \mid (x,y) \in F_{\mathrm{impl}}\} & \text{"post-set in } \Sigma_{\mathrm{impl}}\text{"} \\
\dot{vic}(x) := x^{\blacksquare} \cup {}^{\blacksquare}x & \text{"vicinity in } \Sigma_{\mathrm{impl}}\text{"}
\end{array}
$$

In T_{hotexp}, each transition has its inverse that revokes its effect. The inverse transitions are defined by

$$\bar{t} := \begin{cases} t_{ush} & \text{if } t = t_{vsh} \\ t_{vsh} & \text{if } t = t_{ush} \\ \bar{t}_{fh} & \text{if } t = t_{fh} \\ t_{fh} & \text{if } t = \bar{t}_{fh} \end{cases} \quad \text{for } h \in H,\ s \in vic(h) \quad \text{“inverse transitions in } T_{\text{hotexp}}\text{”.}$$

Of particular significance for modelling Σ_{lat} by Σ_{impl} is the labelling function. It is identity for the transitions from Σ_{lat} (including π), and ignores all others.

$$\alpha(\lambda) := \lambda$$

$$\alpha(wt) := \begin{cases} \alpha(w) & \text{if } t \in T_\tau \\ \alpha(w)t & \text{if } t \in T_v \end{cases} \quad \text{“labelling function”.}$$

Finally, we introduce notions that capture interesting relations in the context of hot expansion.

A latency place that prevents a hot transition from being enabled is said to *obstruct* the hot transition. In ordinary net theory, one can consider an unmarked place in the pre-set and a marked place in the post-set of a transition as ”obstructing” the transition. An enabled transition is not obstructed by any place, but immediately after firing it is obstructed by its entire vicinity. This holds similarly for place latency systems. Here, however, places of the pre-set may obstruct even when they are marked, that is, if the token is latent. In the definition below, *obstr* is introduced as a technical term that does not immediately disclose its relationship to hot transition enabling. That it actually captures the idea is shown in Section 6.5 (Lemma 7).

$$obstr(s,h,c) :\Leftrightarrow \ (s \in {}^\bullet h \wedge w_s \notin c) \vee (s \in h^\bullet \wedge r_s \notin c) \quad \text{“}s \text{ obstructs } h \text{ in } c\text{”.}$$

We say a transition *readies* or *unreadies* a latency place s for a hot transition h if it changes the marking of the representation of s from obstructing h to non-obstructing h, or vice versa, respectively. Technical terms *readies* and *unreadies* are introduced for this idea. In Section 6.5, we show that they actually conform to the idea (Lemmas 8 and 9).

$$\begin{aligned} readies(t,s,h) :\Leftrightarrow\ & h \in H \wedge t \in T_{\text{impl}} \wedge \quad \text{“}t \text{ readies } s \text{ for } h\text{”} \\ & (s \in h^\bullet \cap {}^\bullet t \vee (s \in {}^\bullet h \cap t^\bullet \wedge l(s) = 0) \vee (s \in {}^\bullet h \wedge t = t_{l(s),s})) \end{aligned}$$

$$\begin{aligned} unreadies(t,s,h) :\Leftrightarrow\ & h \in H \wedge t \in T_{\text{lat}} \wedge \\ & (s \in {}^\bullet h \cap {}^\bullet t \vee s \in h^\bullet \cap t^\bullet) \quad \text{“}t \text{ unreadies } s \text{ for } h\text{”.} \end{aligned}$$

Note that *obstr* refers to a marking, while *readies* and *unreadies* are defined in terms of the net structure only. Note also that all three notions refer to Σ_{impl} rather than to

Table 2
Incidence matrix for latency place expansion (associated to $P_{\text{placeexp}} \times T_{\text{placeexp}}$)

		T_{placeexp}				
		q_s	t_{xs}	t_{zs}	t_{is} $i<l(s)$	t_{is} $i=l(s)$
P_{placeexp}	w_s					$+$
	r_s					
	x_s $l(s)>0$	$+$	$+$		$+$	$+$
	y_s $l(s)>0$	$-$			$-$	$-$
	z_s $l(s)>0$	$-$		$+$		$+$
	c_{xs} $l(s)>0$		$-$			
	c_{zs} $l(s)>0$	$+$		$-$		
	c_{is} $1\leqslant i\leqslant l(s)$				$-$	$-$
	$c_{i+1,s}$ $1\leqslant i\leqslant l(s)$				$+$	

Table 3
Incidence matrix for hot expansion (associated to $P_{\text{hotexp}} \times T_{\text{impl}}$)

		T_{lat}			π	T_{placeexp}	T_{hotexp}			
		${}^\bullet s - h$	$s^\bullet - h$	h		$t_{l(s),s}$	t_{ush}	t_{vsh}	t_{fh}	$\bar{t}_{fh}$
P_{hotexp}	dc_{sh} $s\in{}^\bullet h, l(s)=0$	$-$	$+$	$+$			$-$	$+$		
	dc_{sh} $s\in{}^\bullet h, l(s)>0$		$+$	$+$		$-$	$-$	$+$		
	dc_{sh} $s\in h^\bullet$	$+$	$-$	$+$			$-$	$+$		
	f_h				$-$		$+$	$-$	$+$	$-$
	c_{fh}				$+$				$-$	$+$

Table 4
Incidence matrix for latency place linkage (associated to $P_{\text{placeexp}} \times (T_{\text{impl}} - T_{\text{placeexp}})$)

		T_{hotexp}	T_{lat}		π
			${}^\bullet s$	$s^\bullet$	
P_{placeexp}	w_s $l(s)=0$		$+$	$-$	
	r_s $l(s)=0$		$-$	$+$	
	w_s $l(s)>0$			$-$	
	r_s $l(s)>0$		$-$	$+$	
	x_s $l(s)>0$		$-$		$-$
	y_s $l(s)>0$				$+$
	z_s $l(s)>0$		$-$		
	c_{xs} $l(s)>0$		$+$		
	c_{1s} $l(s)>0$		$+$		

Σ_{lat}. Thus, the wordings that a latency place “obstructs”, is “readied” or is “unreadied” are somewhat sloppy. They anticipate that the defined notions actually model features of Σ_{lat} in the intended way.

6.2. Incidence matrix

Tabular descriptions of the net structure are sometimes more convenient than the formulas by which $\Sigma_{\rm impl}$ is defined. In this section, we provide the incidence matrix of the implementation, which is a tabular description of the flow relation $F_{\rm impl}$. Use of the tabular description is formally correct because it represents a set of statements.

Though tables may often be easier to use than formulas, frequent table look-ups may still be somewhat wearisome. Where an intuitive understanding of a proof is deemed sufficient, an inspection of one of the typical net diagrams in Figs. 2 and 5 may be preferable.

To make the incidence matrix manageable, it is broken down into three sub-matrices (Tables 2–4). In their description, s stands for a place from $P_{\rm impl}$, and t stands for a transition from $T_{\rm impl}$. (As usual, s stands for a place of $\Sigma_{\rm lat}$.)

The matrices are parameterized by $s \in P_{\rm lat}$, $h \in H$, and $i > 0$. A field in any of the matrices is applicable to a pair $(\mathsf{s},\mathsf{t}) \in P_{\rm impl} \times T_{\rm impl}$ iff there are parameter values for s, h, and i such that, if the parameters in the row and column heading are given these values, s is in the row heading of the field, t is in (or belongs to the set in) the column heading of the field, and the conditions in the headings are satisfied.

Generally, a given (s,t)-pair uniquely determines its applicable field. To see this, first observe that the domains of the three matrices are exhaustive and mutually exclusive, so that exactly one of the matrices applies. Within the applicable matrix, s and t uniquely select a row and a column, respectively. There are exceptions in the rows for c_{is} and $c_{i+1,s}$ and the columns for q_s, t_{xs} and t_{zs} of the latency place expansion matrix (for these columns, i is not determined by t, so that s may be c_{is} as well as $c_{i+1,s}$), and in the rows for f_h and c_{fh} and the columns for $^{\bullet}s - h$ and $s^{\bullet} - h$ of the hot expansion matrix (for these rows, s is not determined by s, so that t is possibly in the vicinities of several s). However, since all fields concerned are void, no contradiction arises.
$F_{\rm impl}$ is represented in the sense that for $\mathsf{s} \in P_{\rm impl}$ and $\mathsf{t} \in T_{\rm impl}$

$(\mathsf{t},\mathsf{s}) \in F_{\rm impl}$ iff the applicable field for s, t carries a "+"

$(\mathsf{s},\mathsf{t}) \in F_{\rm impl}$ iff the applicable field for s, t carries a "−"

We refer to the sub-matrices by their names rather than by table number.

6.3. Marking representation

In Section 5, the initial marking $c^{\circ}_{\rm impl}$ of $\Sigma_{\rm impl}$ was defined as representing the initial marking $c^{\circ}_{\rm lat}$ of $\Sigma_{\rm lat}$. We are now going to generally map markings of $\Sigma_{\rm impl}$ to markings of $\Sigma_{\rm lat}$.

Of course we require marking representations to conform to their purpose. Not only the initial case, but any case of $\Sigma_{\rm impl}$ must represent a case of $\Sigma_{\rm lat}$. This requires that cases and transition firings in both systems develop in a parallel fashion. We come back to this shortly. Presently, the formal introduction of marking representations just lists them explicitly, together with their respective marking values in $\Sigma_{\rm lat}$.

A marking of $\Sigma_{\rm impl}$ is called *valid* if its intersection with each P_s, $s \in P_{\rm lat}$ is one of the sets in Table 5. Seen the other way round, the first column of Table 5 contains the

Table 5
φ-images

$l(s) > 0$

$m \cap P_s$	$\varphi(m)(s)$	
$\underline{\{x_s, z_s, r_s\}}$ $\{y_s, z_s, r_s\}$ $\{x_s, c_{zs}, r_s\}$ $\{y_s, c_{zs}, r_s\}$	$\circ$	
$\{c_{xs}, c_{1s}\}$ $\underline{\{x_s, c_{1s}\}}$	$l(s)$	
$\underline{\{x_s, c_{is}\}}$ $\{y_s, c_{i-1,s}\}$	$l(s) + 1 - i$	$1 < i \leqslant l(s)$
$\{y_s, c_{l(s),s}\}$ $\underline{\{x_s, z_s, w_s\}}$ $\{y_s, z_s, w_s\}$ $\{x_s, c_{zs}, w_s\}$ $\{y_s, c_{zs}, w_s\}$	$\bullet$	

$l(s) = 0$

$\underline{\{r_s\}}$	$\circ$
$\underline{\{w_s\}}$	$\bullet$

restrictions of all valid markings to a given P_s. The word "valid" is to be understood in the sense that valid markings, and only these, are considered to be legal representations of cases of Σ_{lat}.

Table 5 defines a function φ that associates to each valid marking m of Σ_{impl} and each $s \in P_{\text{lat}}$ a marking value $\varphi(m)(s)$ of Σ_{lat}. As we let s range over P_{lat}, we obtain a marking $\varphi(m)$ of Σ_{lat}. The valid markings of Σ_{impl} that are mapped to the same marking $\varphi(m)$ of Σ_{lat} constitute a *representation class*. The table is grouped according to representation classes.

That the domain and the values of φ conform to the idea of marking representation will be shown in Section 7 by proving that (i) φ and $\triangleright$ commute (Lemma 29), and (ii) the φ-images of cases of Σ_{impl} are cases of Σ_{lat} (Lemmas 30 and 31).

Note that φ does not depend on the marking of the hot expansion P_{hotexp}. This conforms to the intended role of the hot expansion to control enabling of π, which has nothing to do with representing markings of Σ_{lat}.

In the table, the underlined sets are those that belong to the initial case c°_{impl} of Σ_{impl} if the initial case c°_{lat} of Σ_{lat} has the respective marking values in the second column. This can easily be verified by comparing the underlined entries of the table with the definition. From this follow immediately the next two lemmas.

Lemma 2 (Initial case of Σ_{impl} is a valid marking).
c°_{impl} is a valid marking.

Lemma 3 (φ conforms to initial case representation).
$\varphi(c^{\circ}_{\text{impl}}) = c^{\circ}_{\text{lat}}$.

The restrictions of valid markings to P_s are summarized by

Lemma 4 (Restrictions of valid markings to P_s).

$$\begin{aligned}
&\forall m \subseteq P_{\text{impl}}(m \text{ valid marking } \Rightarrow \\
&\forall s \in P^{+}_{\text{lat}}(m \cap P_s \in \{\underline{\{x_s, z_s, r_s\}}, \underline{\{x_s, z_s, w_s\}}, \{x_s, c_{zs}, r_s\}, \{x_s, c_{zs}, w_s\}, \\
&\qquad \{y_s, z_s, r_s\}, \{y_s, z_s, w_s\}, \{y_s, c_{zs}, r_s\}, \{y_s, c_{zs}, w_s\}, \{c_{xs}, c_{1s}\}\} \cup \\
&\qquad \{\underline{\{x_s, c_{is}\}} \mid 1 \leqslant i \leqslant l(s)\} \cup \{\{y_s, c_{is}\} \mid 1 \leqslant i \leqslant l(s)\}) \\
&\wedge \\
&\forall s \in P^{0}_{\text{lat}}(m \cap P_s \in \{\underline{\{r_s\}}, \underline{\{w_s\}}\})).
\end{aligned}$$

The following useful result can also immediately be taken from Table 5.

Lemma 5 (Characteristic properties of φ-values).

$$\begin{aligned}
&\forall s \in P_{\text{lat}},\ m \subseteq P_{\text{impl}}(m \text{ valid marking} \Rightarrow \\
&\quad ((\varphi(m)(s) = \circ \Leftrightarrow r_s \in m) \wedge (\varphi(m)(s) = \bullet \Leftrightarrow w_s \in m \vee m = \{y_s, c_{l(s),s}\}))).
\end{aligned}$$

6.4. Enabledness and follower markings

For convenience, Table 6 is provided as an explicit record of the follower marking for each given valid marking and each transition that is enabled in that marking.

The first column of the table contains all transitions that affect P_s, i.e. those from $v\dot{i}c(P_s) = vic(s) \cup \{\pi\} \cup T_s$. The second column associates restrictions to P_s of valid markings in which each transition is possibly enabled (as far as P_s is concerned). The third column applies only if the transition is enabled. It gives the resulting marking after it has fired. These statements are verified by checking the table row by row with the incidence matrix.

Moreover, the second column completely covers the restrictions to P_s of all valid markings. This is verified by checking the first column of Table 5 row by row with Table 6. Since all sets occurring in the third column also occur in the second column, the table is *closed* in the sense that any set from the third column can be fed back to the second column, and, even more, into the row for any enabled transition affecting

Table 6
Valid markings, enablings, and follower markings

		$m \cap P_s$	$(m \triangleright t) \cap P_s$	
	$l(s) > 0$			
$vic(s)$	$t \in {}^\bullet s$ $t \in s^\bullet$	$\underline{\{x_s, z_s, r_s\}}$ $\underline{\{x_s, z_s, w_s\}}$ $\{y_s, z_s, w_s\}$ $\{x_s, c_{zs}, w_s\}$ $\{y_s, c_{zs}, w_s\}$	$\{c_{xs}, c_{1s}\}$ $\underline{\{x_s, z_s, r_s\}}$ $\{y_s, z_s, r_s\}$ $\{x_s, c_{zs}, r_s\}$ $\{y_s, c_{zs}, r_s\}$	
	π	$\underline{\{x_s, z_s, r_s\}}$ $\{x_s, c_{zs}, r_s\}$ $\underline{\{x_s, c_{is}\}}$ $\underline{\{x_s, z_s, w_s\}}$ $\{x_s, c_{zs}, w_s\}$	$\{y_s, z_s, r_s\}$ $\{y_s, c_{zs}, r_s\}$ $\{y_s, c_{is}\}$ $\{y_s, z_s, w_s\}$ $\{y_s, c_{zs}, w_s\}$	$1 \leqslant i \leqslant l(s)$
T_s	q_s	$\{y_s, z_s, w_s\}$ $\{y_s, z_s, r_s\}$	$\{x_s, c_{zs}, w_s\}$ $\{x_s, c_{zs}, r_s\}$	
T_s	t_{xs}	$\{c_{xs}, c_{1s}\}$	$\{x_s, c_{1s}\}$	
T_s	t_{zs}	$\{x_s, c_{zs}, w_s\}$ $\{y_s, c_{zs}, w_s\}$ $\{x_s, c_{zs}, r_s\}$ $\{y_s, c_{zs}, r_s\}$	$\underline{\{x_s, z_s, w_s\}}$ $\{y_s, z_s, w_s\}$ $\underline{\{x_s, z_s, r_s\}}$ $\{y_s, z_s, r_s\}$	
T_s	t_{is}	$\{y_s, c_{is}\}$	$\{x_s, c_{i+1,s}\}$	$1 \leqslant i < l(s)$
T_s	$t_{l(s),s}$	$\{y_s, c_{l(s),s}\}$	$\underline{\{x_s, z_s, w_s\}}$	
	$l(s) = 0$			
$vic(s)$	$t \in {}^\bullet s$	$\underline{\{r_s\}}$	$\underline{\{w_s\}}$	
$vic(s)$	$t \in s^\bullet$	$\underline{\{w_s\}}$	$\underline{\{r_s\}}$	

P_s. This can be repeated working off an entire firing sequence, considering for each transition the place expansions affected. Thus, starting from a valid marking, only valid markings are reachable. Since we know already that c°_{impl} is a valid marking (Lemma 2), all cases of Σ_{impl} are valid markings.

Lemma 6 (Cases of Σ_{impl} are valid markings).
$\forall c \in C_{\text{impl}}(c \text{ is a valid marking})$.

The converse, that a valid marking is a case, does not hold in general. Since ultimately we are interested in cases only, we will preferably talk about *cases* even if the considerations apply to valid markings as well.

6.5. Hot expansion

Hot expansion is the most subtle part of the implementation. This section shows that it complies with the intuitive idea of controlling π by a counter for the latency places that prevent a hot transition from being enabled. For this, the notions of *obstr*, *readies* and *unreadies* play a dominant role. Lemmas 7–10 show that these concepts conform to the intuition. Lemmas 11–18 prepare the proofs of Lemmas 19–21, which are the main results of this section. They show that the hot expansion correctly represents the deficiency count.

In this section, we have to talk about vicinities both in Σ_{lat} and Σ_{impl}. In formal expressions they are clearly distinguished by the use of *vic* or $\dot{vic}$, respectively. To avoid confusion in text, we use vicinity$_L$ for referring to Σ_{lat}, and vicinity$_I$ for referring to Σ_{impl}. This notation is used in the present section only.

Lemma 7 (Obstruction for enabled hot transition).
$$\forall h \in H, \quad c \in C_{\text{impl}}(c \triangleright h \Rightarrow \{s \mid obstr(s,h,c)\} = \emptyset \wedge \{s \mid obstr(s,h,c \triangleright h)\} = vic(h)).$$

Proof. Let $s \in {}^{\bullet}h$. By the incidence matrix for latency place linkage we have $w_s \in {}^{\bullet}h$, hence, because of $c \triangleright h$, holds $w_s \in c$ and $w_s \notin c \triangleright h$. Similarly, for $s \in h^{\bullet}$ follows $r_s \in c$ and $r_s \notin c \triangleright h$. By the definition of *obstr* then holds $\neg\, obstr(s,h,c)$ and $obstr(s,h,c \triangleright h)$ in both cases. For $s \notin vic(h)$ is $\neg\, obstr(s,h,c)$ regardless of c. Thus, for any $c \in C_{\text{impl}}$, we have $\neg\, obstr(s,h,c)$ and $obstr(s,h,c \triangleright h) \Leftrightarrow s \in vic(h)$. $\square$

The lemma asserts that *obstr* complies with what we had in mind when we introduced the concept (see Section 6.1).

The next two lemmas justify the wording “t readies (unreadies) s for h” for characterizing a particular incidence situation between a transition, a place, and a hot transition. It says that, in such a situation, firing the enabled t changes s from obstructing h to non-obstructing h (or vice versa, correspondingly).

Lemma 8 (*readies* and *obstr*).
$$\forall h \in H, \quad t \in T_{\text{impl}}, \quad s \in P_{\text{lat}}, \quad c \in C_{\text{impl}}$$
$$(c \triangleright t \wedge readies(t,s,h) \Rightarrow obstr(s,h,c) \wedge \neg\, obstr(s,h,c \triangleright t)).$$

Proof. The three alternatives in the definition of *readies* afford identical reasoning for showing the claimed properties. It is therefore convenient to arrange the arguments in

Table 7
$obstr(s,h,c)$ and $\neg\, obstr(s,h,c \triangleright t)$ follow from $readies(t,s,h)$

Alternative of *readies*	$s \in h^{\bullet} \cap {}^{\bullet}t$	$s \in {}^{\bullet}h \cap t^{\bullet},\ l(s)=0$	$s \in {}^{\bullet}h,\ t = t_{l(s),s}$
Net-theoretic consequences	$t \in s^{\bullet},\ s \notin {}^{\bullet}h$	$t \in {}^{\bullet}s,\ s \notin h^{\bullet}$	$s \in {}^{\bullet}h,\ s \notin h^{\bullet}$
Incidence matrix for lat. place	Linkage	Linkage	Expansion
yields	$r_s \in t^{\blacksquare}$	$w_s \in t^{\blacksquare}$	$w_s \in t^{\blacksquare}$
$c \triangleright t$ entails	$r_s \notin c,\ r_s \in c \triangleright t$	$w_s \notin c,\ w_s \in c \triangleright t$	$w_s \notin c,\ w_s \in c \triangleright t$
Summary for $obstr(s,h,c)$	$s \in h^{\bullet},\ r_s \notin c$	$s \in {}^{\bullet}h,\ w_s \notin c$	$s \in {}^{\bullet}h,\ w_s \notin c$
Summary for $\neg\, obstr(s,h,c \triangleright t)$	$s \notin {}^{\bullet}h,\ r_s \in c \triangleright t$	$s \notin h^{\bullet},\ w_s \in c \triangleright t$	$s \notin h^{\bullet},\ w_s \in c \triangleright t$

Table 8
$\neg\, obstr(s,h,c)$ and $obstr(s,h,c \triangleright t)$ follow from $unreadies(t,s,h)$

Alternative of *unreadies*	$s \in {}^{\bullet}h \cap {}^{\bullet}t$	$s \in h^{\bullet} \cap t^{\bullet}$
Net-theoretic consequences	$t \in s^{\bullet},\ s \notin h^{\bullet}$	$t \in {}^{\bullet}s,\ s \notin {}^{\bullet}h$
Incidence matrix for lat. place	Linkage	Linkage
yields	$w_s \in {}^{\blacksquare}t$	$r_s \in {}^{\blacksquare}t$
$c \triangleright t$ entails	$w_s \in c,\ w_s \notin c \triangleright t$	$r_s \in c,\ r_s \notin c \triangleright t$
Summary for $\neg\, obstr(s,h,c)$	$s \notin h^{\bullet},\ w_s \in c$	$s \notin {}^{\bullet}h,\ r_s \in c$
Summary for $obstr(s,h,c \triangleright t)$	$s \in {}^{\bullet}h,\ w_s \notin c \triangleright t$	$s \in h^{\bullet},\ r_s \notin c \triangleright t$

a schematic way as given in Table 7. By the definition of *obstr*, the summary rows give the desired results. □

Lemma 9 (*unreadies* and *obstr*).

$$\forall h \in H,\ t \in T_{\text{impl}},\ s \in P_{\text{lat}},\ c \in C_{\text{impl}}$$
$$(c \triangleright t \wedge unreadies(t,s,h) \Rightarrow \neg\, obstr(s,h,c) \wedge obstr(s,h,c \triangleright t)).$$

Proof. The proof follows the same lines as that for Lemma 8. It is simpler though, since the definition of *unreadies* has only two alternatives rather than three. Table 8 gives the desired result. □

Readies and *unreadies* exclude each other, but they are not complementary since there are incidence situations not covered by either of them. For these, firing the transition does not alter obstruction. This is expressed by the next two lemmas.

Lemma 10 (*readies* and *unreadies* exclude each other).

$$\forall h \in H, \quad t \in T_{\text{impl}}, \quad s \in P_{\text{lat}}(\neg(readies(t,s,h) \wedge unreadies(t,s,h))).$$

Proof. From $readies(t,s,h) \wedge unreadies(t,s,h)$ would follow the weaker formula

$$(s \in h^{\bullet} \cap {}^{\bullet}t \vee s \in {}^{\bullet}h \cap t^{\bullet}) \wedge (s \in {}^{\bullet}h \cap {}^{\bullet}t \vee s \in h^{\bullet} \cap t^{\bullet})$$

by straightforward propositional logic calculation. But this is contradictory because nets are required to be pure. □

Lemma 11 (Relations between *readies*, *unreadies* and *obstr*).
$\forall t \in T_{\mathrm{impl}},\ s \in P_{\mathrm{lat}},\ h \in H,\ c \in C_{\mathrm{impl}}$
$(c \triangleright t \Rightarrow readies(t,s,h) \vee unreadies(t,s,h) \vee (obstr(s,h,c) \Leftrightarrow obstr(s,h,c \triangleright t)))$.

Proof. We show that, if the equivalence $obstr(s,h,c) \Leftrightarrow obstr(s,h,c \triangleright t)$ does not hold, either $readies(t,s,h)$ or $unreadies(t,s,h)$. The negation of the equivalence can be expressed

$$(obstr(s,h,c) \wedge \neg obstr(s,h,c \triangleright t)) \vee (\neg obstr(s,h,c) \wedge obstr(s,h,c \triangleright t)).$$

For both alternatives holds $s \in vic(h)$, i.e., $s \in {}^{\bullet}h$ or $s \in h^{\bullet}$. Combining the two pairs of alternatives, we arrive at four combinations which we consider separately.

For $obstr(s,h,c) \wedge \neg obstr(s,h,c \triangleright t)$ and $s \in {}^{\bullet}h$ the definition of *obstr* yields $w_s \notin c$ and $w_s \in c \triangleright t$. That is, t puts a token onto w_s. The incidence matrices for latency place expansion and linkage then say $t \in {}^{\bullet}s \wedge l(s) = 0$ or $t = t_{l(s),s}$. From both alternatives follows $readies(t,s,h)$ by the definition of *readies*.

For $obstr(s,h,c) \wedge \neg obstr(s,h,c \triangleright t)$ and $s \in h^{\bullet}$ the definition of *obstr* yields $r_s \notin c$ and $r_s \in c \triangleright t$. That is, t puts a token onto r_s. The incidence matrices for latency place expansion and linkage then say $t \in s^{\bullet}$. From this follows $readies(t,s,h)$ by the definition of *readies*.

For $\neg obstr(s,h,c) \wedge obstr(s,h,c \triangleright t)$ and $s \in {}^{\bullet}h$ the definition of *obstr* yields $w_s \in c$ and $w_s \notin c \triangleright t$. That is, t removes a token from w_s. The incidence matrices for latency place expansion and linkage then say $t \in s^{\bullet}$. From this follows $unreadies(t,s,h)$ by the definition of *unreadies*.

For $\neg obstr(s,h,c) \wedge obstr(s,h,c \triangleright t)$ and $s \in h^{\bullet}$ the definition of *obstr* yields $r_s \in c$ and $r_s \notin c \triangleright t$. That is, t removes a token from r_s. The incidence matrices for latency place expansion and linkage then say $t \in {}^{\bullet}s$. From this follows $unreadies(t,s,h)$ by the definition of *unreadies*. $\square$

We have seen that an enabled hot transition is not obstructed, but after it has fired, it is obstructed by all of its vicinity$_L$ (Lemma 7). That is, its firing changes the vicinity$_L$ from not obstructing to obstructing. Thus one should expect that a hot transition unreadies its vicinity$_L$ for itself (and does, of course, not ready any latency place for itself). This is expressed by

Lemma 12 (Hot transition unreadies its vicinity$_L$ for itself).
$\forall h \in H(\{s \mid readies(h,s,h)\} = \emptyset \wedge \{s \mid unreadies(h,s,h)\} = vic(h))$.

Proof. For a hot transition h, $unreadies(h,s,h)$ is equivalent to $s \in {}^{\bullet}h \vee s \in h^{\bullet}$ by definition, i.e., $s \in vic(h)$. This accounts for $\{s \mid unreadies(h,s,h)\} = vic(h)$. From this and Lemma 10 follows $\neg readies(h,s,h)$ for $s \in vic(h)$. Since the definition of *readies* requires $s \in vic(h)$, we have $\neg readies(h,s,h)$ also for $s \notin vic(h)$. Thus $\{s \mid readies(h,s,h)\} = \emptyset$. $\square$

readies and *unreadies* are defined in terms of incidence relations in Σ_{lat}. There is also a convenient characterization of such a situation in Σ_{impl}:

Lemma 13 (Characterization of *readies* and *unreadies* in Σ_{impl}).
$\forall h \in H, \quad t \in T_{\mathrm{impl}} - T_{\mathrm{hotexp}}$
$(\{s \mid readies(t,s,h)\} = \{s \mid dc_{sh} \in {}^{\bullet}t\} \wedge \{s \mid unreadies(t,s,h)\} = \{s \mid dc_{sh} \in t^{\bullet}\})$.

Proof. For $s \notin vic(h)$, the place dc_{sh} is undefined. For $s \in vic(h)$, the incidence matrix for hot expansion says that $dc_{sh} \notin {}^{\bullet}h$ and $dc_{sh} \in h^{\bullet}$. Hence $\{s \mid dc_{sh} \in {}^{\bullet}h\} = \emptyset$ and $\{s \mid dc_{sh} \in h^{\bullet}\} = vic(h)$. For $t = h$, the claimed equations then follow from Lemma 12.

Let now $t \neq h$. Since $t \notin T_{\mathrm{hotexp}}$, the incidence matrix for hot expansion yields

$$dc_{sh} \in {}^{\bullet}t \Leftrightarrow (s \in h^{\bullet} \wedge t \in s^{\bullet}) \vee (s \in {}^{\bullet}h \wedge t \in {}^{\bullet}s \wedge l(s) = 0) \vee (s \in {}^{\bullet}h \wedge t = t_{l(s),s})$$

by its rows for dc_{sh}, and its columns for $s^{\bullet} - h, {}^{\bullet}s - h$ and $t_{l(s),s}$,

$$dc_{sh} \in t^{\bullet} \Leftrightarrow (s \in {}^{\bullet}h \wedge t \in s^{\bullet}) \vee (s \in h^{\bullet} \wedge t \in {}^{\bullet}s)$$

by its rows for dc_{sh}, and its columns for $s^{\bullet} - h$ and ${}^{\bullet}s - h$.

(Note that in the last term of the first line $l(s) > 0$ is justly omitted because it is implied by $t = t_{l(s),s}$.)

By the definitions of *readies* and *unreadies* follows

$$dc_{sh} \in {}^{\bullet}t \Leftrightarrow readies(t,s,h),$$

$$dc_{sh} \in t^{\bullet} \Leftrightarrow unreadies(t,s,h). \qquad \square$$

We are now approaching the central issue of this section, the relation between the cardinality of a hot expansion marking and the number of obstructing places, i.e., the deficiency count. Lemmas 14–17 deal with particular cardinality properties needed for subsequent proofs.

Lemma 14 (Correspondence between latency places and representation in P_{hotexp}).
$\forall h \in H, \quad P \subseteq P_{\mathrm{impl}}(\|\{s \mid dc_{sh} \in P\}\| = \|\{dc_{sh} \mid dc_{sh} \in P\}\|)$.

Proof. This is an immediate consequence of our convention that different symbols mean different objects (see Section 5). Thus $dc_{sh} \neq dc_{s'h}$ for $s \neq s'$, so that there is a one–one correspondence between the s and the dc_{sh}. (Note that $\{dc_{sh} \mid dc_{sh} \in P\}$ is not the same as P, since P may contain other elements from P_{impl}.) $\square$

Lemma 15 (Correspondence between readying/unreadying and vicinity$_I$ within P_{hotexp}).
$\forall h \in H, \quad t \in T_{\mathrm{impl}} - T_{\mathrm{hotexp}} - \{\pi\}$
$(\|\{s \mid readies(t,s,h)\}\| = \|{}^{\bullet}t \cap P_h\| \wedge \|\{s \mid unreadies(t,s,h)\}\| = \|t^{\bullet} \cap P_h\|)$.

Proof. In the applicable columns (for T_{lat} and T_{placeexp}) of the incidence matrix for hot expansion, the entries for f_h and c_{fh} are void. Hence elements of ${}^{\bullet}t \cap P_h$ and $t^{\bullet} \cap P_h$

must be some dc_{sh}. Thus

$$^{\bullet}t \cap P_h = \{dc_{sh} \mid dc_{sh} \in {}^{\bullet}t\} \quad \text{and} \quad t^{\bullet} \cap P_h = \{dc_{sh} \mid dc_{sh} \in t^{\bullet}\}.$$

From Lemma 13, we have

$$\{s \mid readies(t,s,h)\} = \{s \mid dc_{sh} \in {}^{\bullet}t\} \text{ and } \{s \mid unreadies(t,s,h)\} = \{s \mid dc_{sh} \in t^{\bullet}\}.$$

Linking these equations by Lemma 14 [${}^{\bullet}t$ for P] or [$t^{\bullet}$ for P], respectively, we obtain the desired result. □

Lemma 16 (Cardinality change on hot expansion marking by transition firing).

$$\forall h \in H,\ t \in T_{\text{impl}},\ c \in C_{\text{impl}}$$
$$(c \triangleright t \Rightarrow \|c \triangleright t \cap P_h\| = \|c \cap P_h\| - \|\{s \mid readies(t,s,h)\}\| + \|\{s \mid unreadies(t,s,h)\}\|).$$

Proof. If $t \in T_{\text{impl}} - T_{\text{lat}} - T_{\text{exit}}$, then $\neg\, readies(t,s,h)$ and $\neg\, unreadies(t,s,h)$ follow for any $s \in P_{\text{lat}}$ because the definition of *readies* and *unreadies* requires $t \in vic(s)$ and therefore $t \in T_{\text{lat}}$, or $t = t_{l(s),s}$ and therefore $t \in T_{\text{exit}}$. Thus, $\{s \mid readies(t,s,h)\} = \{s \mid unreadies(t,s,h)\} = \emptyset$. Further can be seen from the incidence matrix for hot expansion that $\|{}^{\bullet}t \cap P_h\| = \|t^{\bullet} \cap P_h\|$. Hence we have $\|\{s \mid readies(t,s,h)\}\| - \|\{s \mid unreadies(t,s,h)\}\| = \|{}^{\bullet}t \cap P_h\| - \|t^{\bullet} \cap P_h\| = 0$.

If $t \in T_{\text{lat}} \cup T_{\text{exit}}$, then by definition of T_{impl} is $t \notin T_{\text{hotexp}} \cup \{\pi\}$. This allows to apply Lemma 15 to obtain $\|{}^{\bullet}t \cap P_h\| = \|\{s \mid readies(t,s,h)\}\|$ and $\|t^{\bullet} \cap P_h\| = \|\{s \mid, unreadies(t,s,h)\}\|$.

By (Net2) [c for $m; P_h$ for d] we have $\|c \triangleright t \cap P_h\| = \|c \cap P_h\| - \|{}^{\bullet}t \cap P_h\| + \|t^{\bullet} \cap P_h\|$. From this and the above partial results follows the claimed property. □

Lemma 17 (Obstruction change by transition firing).

$$\forall h \in H,\ t \in T_{\text{impl}},\ c \in C_{\text{impl}}$$
$$(c \triangleright t \Rightarrow \{s \mid obstr(s,h,c \triangleright t)\} =$$
$$(\{s \mid obstr(s,h,c)\} - \{s \mid readies(t,s,h)\}) \cup \{s \mid unreadies(t,s,h)\}).$$

Proof. If $t = h$, then $\{s \mid obstr(s,h,c \triangleright t)\} = vic(h)$ and $\{s \mid obstr(s,h,c)\} = \emptyset$ according to Lemma 7. Lemma 12 yields $\{s \mid readies(t,s,h)\} = \emptyset$ and $\{s \mid unreadies(t,s,h)\} = vic(h)$. Together this makes the claimed property.

Let now $t \neq h$ and $s \in P_{\text{lat}}$. From $readies(t,s,h)$ follows $\neg\, obstr(s,h,c \triangleright t)$ by Lemma 8, and $\neg\, unreadies(t,s,h)$ by Lemma 10. From $unreadies(t,s,h)$ follows $obstr(s,h,c \triangleright t)$ by Lemma 9. If $\neg\, readies(t,s,h)$ and $\neg\, unreadies(t,s,h)$, then $obstr(s,h,c) \Leftrightarrow obstr(s,h,c \triangleright t)$ by Lemma 11. Hence $obstr(s,h,c \triangleright t) \Leftrightarrow (obstr(s,h,c) \wedge \neg\, readies(t,s,h)) \vee unreadies(t,s,h)$ holds in all three cases. From this follows the claimed set equation. □

The next two lemmas deal with the "deficiency count property", that is, with the idea that the expansion for a single hot transition acts as a counter for the latency places that obstruct the hot transition. Lemma 18 states the deficiency count property for the initial case. With Lemma 19 it is shown that this property is inherited to a follower case, so that it is true for any case.

Lemma 18 (Deficiency count property for initial case).
$\forall h \in H \ (\|c^{\circ}_{\mathrm{impl}} \cap P_h\| = \|\{s \mid obstr(s,h,c^{\circ}_{\mathrm{impl}})\}\|)$.

Proof. For $c^{\circ}_{\mathrm{impl}} \cap P_h$, the first five sets of the definition of $c^{\circ}_{\mathrm{impl}}$ do not contribute anything because they do not belong to P_{hotexp}. The last three sets yield

$$\begin{aligned} c^{\circ}_{\mathrm{impl}} \cap P_h = \{dc_{sh} \mid\ & (c^{\circ}_{\mathrm{lat}}(s) = \circ \wedge s \in {}^{\bullet}h) \vee \\ & (c^{\circ}_{\mathrm{lat}}(s) = \bullet \wedge s \in h^{\bullet}) \vee \\ & (c^{\circ}_{\mathrm{lat}}(s) \notin \{\circ, \bullet\} \wedge s \in vic(h))\}. \end{aligned}$$

Using Lemma 1 and splitting up $vic(h)$, we obtain

$$\begin{aligned} c^{\circ}_{\mathrm{impl}} \cap P_h = \{dc_{sh} \mid\ & (r_s \in c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h) \vee \\ & (w_s \in c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet}) \vee \\ & (r_s, w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h) \vee \\ & (r_s, w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet})\}. \end{aligned}$$

The partial formula $(r_s \in c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h) \vee (r_s, w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h)$ is logically equivalent to $(r_s \in c^{\circ}_{\mathrm{impl}} \vee w_s \notin c^{\circ}_{\mathrm{impl}}) \wedge s \in {}^{\bullet}h$. Using $r_s \notin c^{\circ}_{\mathrm{impl}} \vee w_s \notin c^{\circ}_{\mathrm{impl}}$ from Lemma 1, this is in turn equivalent to $w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h$. Correspondingly, $(w_s \in c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet}) \vee (r_s, w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet})$ is equivalent to $r_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet}$. Together this results in

$$c^{\circ}_{\mathrm{impl}} \cap P_h = \{dc_{sh} \mid (w_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in {}^{\bullet}h) \vee (r_s \notin c^{\circ}_{\mathrm{impl}} \wedge s \in h^{\bullet})\},$$

or

$$c^{\circ}_{\mathrm{impl}} \cap P_h = \{dc_{sh} \mid obstr(s,h,c^{\circ}_{\mathrm{impl}})\}$$

by the definition of *obstr*. Since by convention $dc_{sh} \neq dc_{s'h}$ for $s \neq s'$, we obtain

$$\|c^{\circ}_{\mathrm{impl}} \cap P_h\| = \|\{dc_{sh} \mid obstr(s,h,c^{\circ}_{\mathrm{impl}})\}\| = \|\{s \mid obstr(s,h,c^{\circ}_{\mathrm{impl}})\}\|. \qquad \square$$

Lemma 19 (Deficiency count property).
$\forall h \in H, \quad c \in C_{\mathrm{impl}}(\|c \cap P_h\| = \|\{s \mid obstr(s,h,c)\}\|)$.

Proof. The case c is reached from $c^{\circ}_{\mathrm{impl}}$ by some firing sequence. We prove the lemma by recursion over the minimal possible length of a firing sequence w such that $c = c^{\circ}_{\mathrm{impl}} \triangleright w$.

If this length is 0, then $w = \lambda$ and $c = c^{\circ}_{\mathrm{impl}}$. Lemma 18 provides the desired result. Let now the minimal possible length be positive with $w = w't$ and $c = c^{\circ}_{\mathrm{impl}} \triangleright w't$. The case $c' = c^{\circ}_{\mathrm{impl}} \triangleright w'$ is reached by a shorter firing sequence w', so that by the recursion assumption $\|c' \cap P_h\| = \|\{s \mid obstr(s,h,c')\}\|$. Lemma 17 [$c'$ for c] yields

$$\{s \mid obstr(s,h,c)\} = (\{s \mid obstr(s,h,c')\} - \{s \mid readies(t,s,h)\}) \\ \cup \{s \mid unreadies(t,s,h)\}.$$

The subtracted set is contained in $\{s \mid obstr(s,h,c')\}$ by Lemma 8 [c' for c]. The added set is disjoint to $\{s \mid obstr(s,h,c')\}$ by Lemma 9 [c' for c]. Thus, for set-theoretic reasons, the set-operations are mirrored by the cardinalities, i.e.,

$$\begin{aligned} &\|\{s \mid obstr(s,h,c)\}\| \\ &\quad = \|\{s \mid obstr(s,h,c')\}\| - \|\{s \mid readies(t,s,h)\}\| + \|\{s \mid unreadies(t,s,h)\}\|. \end{aligned}$$

By Lemma 16 [c' for c] we have

$$\|c' \triangleright t \cap P_h\| = \|c' \cap P_h\| - \|\{s \mid readies(t,s,h)\}\| + \|\{s \mid unreadies(t,s,h)\}\|,$$

which together with $c' \triangleright t = c$ and $\|c' \cap P_h\| = \|\{s \mid obstr(s,h,c')\}\|$ yields the desired result. □

Next, the expression that tokens "may float freely" on the expansion of a single hot transition (see *Hot expansion* in Section 5) is given the formal meaning that a given number of tokens may be shuffled by τ-transitions to any given distribution on the hot expansion.

Lemma 20 (Tokens may float freely on a hot expansion).

$$\forall m, m' \subseteq P_{\text{impl}}, \quad H' \subseteq H \\ (m \ominus m' \subseteq P_{H'} \wedge \forall h \in H'(\|m \cap P_h\| = \|m' \cap P_h\|) \Rightarrow \exists w \in T^*_{H'}(m' = m \triangleright w)).$$

Proof. By recursion over $\|m \ominus m'\|$.

If $\|m \ominus m'\| = 0$, then $m = m'$ and $w := \lambda$ trivially has the desired property. If $\|m \ominus m'\| > 0$, then $m \ominus m' \neq \emptyset$ and, because of $m \ominus m' \subseteq P_{H'}$, there must be an $h \in H'$ such that $(m \ominus m') \cap P_h \neq \emptyset$. Thus, at least one of $(m - m') \cap P_h$ or $(m' - m) \cap P_h$ must be non-empty. Actually, both must be non-empty because they have the same cardinality. This follows by set-theoretic considerations from $\|m \cap P_h\| = \|m' \cap P_h\|$, which is an assumption of the lemma. Assume $p_1 \in (m - m') \cap P_h$ and $p_2 \in (m' - m) \cap P_h$, $p_1 \neq p_2$, to be elements by which m and m' differ. We can undo the particular deviation of m' from m caused by these elements by defining $m'' := (m' - \{p_2\}) \cup \{p_1\}$, thus obtaining a smaller symmetric difference than for m': $\|m \ominus m''\| = \|m \ominus m'\| - 2 < \|m \ominus m'\|$. This allows to apply the recursion assumption. We have only to verify that the assumptions of the lemma hold for m''. Obviously, $m \ominus m'' \subseteq P_{H'}$ because $p_1, p_2 \in P_h$. Further, by its definition, m'' differs from m' only on P_h, hence $m'' \cap P_{h'} = m' \cap P_{h'}$ holds for $h' \in H - h$, all the more $\|m \cap P_{h'}\| = \|m'' \cap P_{h'}\|$. Moreover, the difference results from discarding an element from $m' \cap P_h$ and adding an element not from $m' \cap P_h$, so that the cardinality on P_h is unchanged, i.e., $\|m'' \cap P_h\| = \|m' \cap P_h\|$. Thus, the recursion assumption yields a $w'' \in T^*_{H'}$ such that $m'' = m \triangleright w''$. Table 9 surveys the possible combinations of p_1 and

Table 9
Building a deficiency count shuffle sequence

			$p_2 =$			
			c_{fh}	f_h	dc_{s_2h}	
					$f_h \in m''$	$f_h \notin m''$
$p1 =$	c_{fh}		—	t_{fh}	$t_{vs_2h}t_{fh}$	$t_{fh}t_{vs_2h}$
	f_h		$\bar{t}_{fh}$	—	t_{vs_2h}	—
	dc_{s_1h}	$f_h \in m''$	$\bar{t}_{fh}t_{us_1h}$	—	$t_{vs_2h}t_{us_1h}$	—
		$f_h \notin m''$	$t_{us_1h}\bar{t}_{fh}$	t_{us_1h}	—	$t_{us_1h}t_{vs_2h}$

p_2 and associates sequences that will be used to construct the desired w. Note that, whenever s_1 and s_2 occur together, $s_1 \neq s_2$ because $p_1 \neq p_2$.

In the diagram, the dash ("—") indicates excluded combinations of values for p_1 and p_2. The two excluded combinations in the upper left part follow from $p_1 \neq p_2$. The exclusion for row $p_1 = f_h$, column $f_h \notin m''$ follows from $p_1 \in m''$. Similarly, $p_2 \notin m''$ accounts for the exclusion of row $f_h \in m''$, column $p_2 = f_h$. The remaining two exclusions are due to contradictions among the row and column headers.

The desired sequence is now constructed as $w := w''w'$, where w' is the sequence associated to given p_1, p_2 by Table 9. For each combination of p_1 and p_2 it is easy to establish that (i) $m'' \triangleright w'$ and (ii) $m'' \triangleright w' = (m'' - p_1) \cup p_2$, using $p_1 \in m''$, $p_2 \notin m''$ and relevant vicinity$_I$ information taken directly from the incidence matrix for hot expansion:

$$\begin{array}{llll} {}^\bullet t_{fh} = c_{fh} & {}^\bullet \bar{t}_{fh} = f_h & {}^\bullet t_{us_1h} = dc_{s_1h} & {}^\bullet t_{vs_2h} = f_h \\ t_{fh}{}^\bullet = f_h & \bar{t}_{fh}{}^\bullet = c_{fh} & t_{us_1h}{}^\bullet = f_h & t_{vs_2h}{}^\bullet = dc_{s_2h}. \end{array}$$

From $m'' = (m' - \{p_2\}) \cup \{p_1\}$ with $p_1 \notin m'$, $p_2 \in m'$ follows $m' = (m'' - \{p_1\}) \cup \{p_2\}$. By this and (ii) we have $m' = m'' \triangleright w' = m \triangleright w''w'$. Since all possible w' are from $T_h^* \subseteq T_{H'}^*$ and $w'' \in T_{H'}^*$ we have also $w \in T_{H'}^*$. $\square$

With the help of the last two lemmas it is easy to show that, if no hot transition is enabled, the tokens on the hot expansion can be shuffled by τ-transitions such that π is not impeded by the hot expansion.

Lemma 21 (Enabling π on hot expansion).
$\forall c \in C_{\text{impl}}(\forall h \in H(\neg c \triangleright h) \Rightarrow \exists w \in T^*_{\text{hotexp}}\ \forall h \in H(f_h \in c \triangleright w \ \wedge \ c_{fh} \notin c \triangleright w))$.

Proof. Let $h \in H$. Since $obstr(s,h,c)$ implies $s \in vic(h)$ by the definition of *obstr*, we have $\|\{s \mid obstr(s,h,c)\}\| \leqslant \|vic(h)\|$, or, with Lemma 19, $\|c \cap P_h\| \leqslant \|vic(h)\|$. The definition of P_h yields $\|P_h\| = \|vic(h)\| + 2$, so that $\|c \cap P_h\| < \|P_h\|$. Thus, there are at least one marked and one unmarked place in P_h. Therefore, the tokens on P_h can be re-distributed so that f_h is marked and c_{fh} is unmarked. Let c' be a marking obtained from c by shuffling the tokens in this sense on all the P_h, that is, $c' \cap P_{\text{placeexp}} = c \cap P_{\text{placeexp}}$, and $f_h \in c'$, $c_{fh} \notin c'$, $\|c' \cap P_h\| = \|c \cap P_h\|$ for all $h \in H$. By Lemma 20 [c for m; c' for m'; H for H'], we obtain a $w \in T^*_{\text{hotexp}}$ such that $c' = c \triangleright w$ which, by construction of c', has the desired properties. □

6.6. Hot normal and emergent cases

According to Lemma 20, tokens on each P_h, $h \in H$, can arbitrarily be shuffled around by τ-transitions. Their purpose however, namely to represent the deficiency count for h, is best visualized by the particular distribution of tokens where the marking of each dc_{sh} expresses the impact of the expansion of an $s \in vic(h)$ on the enabling of h, that is, if exactly those dc_{sh} are marked for which holds $obstr(s,h,c)$. Cases with this property are called *hot normal*. They are useful because (i) a hot normal case can always be reached by τ-transitions (Lemma 22), and (ii) in a hot normal case it is sufficient to inspect the markings of latency place expansions to determine whether a visible proper transition is enabled (Lemma 23).
c hot normal $:\Leftrightarrow \forall s \in P_{\text{impl}},\ h \in H(obstr(s,h,c) \Leftrightarrow dc_{sh} \in c)$.

Lemma 22 (Reaching a hot normal case).
$\forall c \in C_{\text{impl}}\ \exists\, w \in T^*_{\text{hotexp}}\ (c \triangleright w$ hot normal$)$.

Proof. We define $c' := (c \cap P_{\text{placeexp}}) \cup \{dc_{sh} \mid obstr(s,h,c)\}$, that is, c' is identical to c on P_{placeexp} and has tokens in P_{hotexp} exactly on the places dc_{sh} with $obstr(s,h,c)$. For any $h \in H$ follows $c' \cap P_h = \{dc_{sh} \mid obstr(s,h,c)\}$ since $P_{\text{placeexp}} \cap P_h = \emptyset$ and $dc_{sh} \in P_h$ by definition of P_h. Remembering that different symbols mean different objects, this yields $\|c' \cap P_h\| = \|\{s \mid obstr(s,h,c)\}\|$, and, with Lemma 19, $\|c' \cap P_h\| = \|c \cap P_h\|$. This holds for all $h \in H$, so that c' is just a re-shuffling of tokens on P_{hotexp}. By Lemma 20 [c for m; c' for m'; H for H'] we obtain a $w \in T^*_{\text{hotexp}}$ such that $c' = c \triangleright w$. By construction of c', w has the desired properties.

Lemma 23 (Enabling in hot normal case).
$\forall t \in T_{\text{lat}},\quad c \in C_{\text{impl}}(c$ hot normal $\wedge\ {}^\bullet t \cap P_{\text{placeexp}} \subseteq c \wedge t^\bullet \cap P_{\text{placeexp}} \cap c = \emptyset \Rightarrow c \triangleright t)$.

Proof. First, the pre-set and the post-set of t are decomposed into their latency place expansion and their hot expansion part (see definition of P_{impl}), that is, ${}^\bullet t = ({}^\bullet t \cap P_{\text{placeexp}}) \cup ({}^\bullet t \cap P_{\text{hotexp}})$ and $t^\bullet = (t^\bullet \cap P_{\text{placeexp}}) \cup (t^\bullet \cap P_{\text{hotexp}})$. To prove $c \triangleright t$, we have to establish ${}^\bullet t \subseteq c\ \wedge\ t^\bullet \cap c = \emptyset$. Since ${}^\bullet t \cap P_{\text{placeexp}} \subseteq c$ and $t^\bullet \cap P_{\text{placeexp}} \cap c = \emptyset$ are already presup-

Table 10
Vicinity in hot expansion

$s\in$	$\notin c$	$dc_{sh}\in c$
$h^\bullet \cap {}^\bullet t$	r_s	yes
${}^\bullet h\cap t^\bullet, l(s)=0$	w_s	yes
${}^\bullet h\cap {}^\bullet t, h\neq t$	r_s	no
$h^\bullet \cap t^\bullet, h\neq t$	w_s	no
${}^\bullet t, h=t$	r_s	no
$t^\bullet, h=t$	w_s	no

posed, it remains to be shown that ${}^\bullet t\cap P_{\text{hotexp}}\subseteq c$ and $t^\bullet\cap P_{\text{hotexp}}\cap c=\emptyset$, or, equivalently

$(*)\quad {}^\bullet t\cap P_h\subseteq c$ and $t^\bullet\cap P_h\cap c=\emptyset \quad$ for all $h\in H$.

Let $h\in H$. Table 10 summarizes some information needed in this proof. It is valid under the assumptions of the lemma only.

To construct Table 10, let s be in some set in column 1. The incidence matrix for latency place linkage says that for $t\in s^\bullet$ follows $r_s\in t^\bullet$. With $t^\bullet\cap P_{\text{placeexp}}\cap c=\emptyset$ this yields $r_s\notin c$. Similarly, follows $r_s\in{}^\bullet t$ from $t\in{}^\bullet s$, and $r_s\in c$ with ${}^\bullet t\cap P_{\text{placeexp}}\subseteq c$. Lemma 4 then yields $w_s\notin c$. This together makes column 2. From the definition of *obstr*, applied to the first two columns, we get $obstr(s,h,c)$ for the upper two rows, hence, since c is hot normal, $dc_{sh}\in c$. Similarly, for the lower four rows we obtain $\neg\ obstr(s,h,c)$, hence $dc_{sh}\notin c$. This is noted in column 3.

If $v\dot{i}c(t)\cap P_h=\emptyset$, then $(*)$ is trivially true. Let $p\in v\dot{i}c(t)\cap P_h$. Then one of the non-void entries in the columns for T_{lat} of the incidence matrix for hot expansion applies, and there must be an $s\in P_{\text{lat}}$ such that $p=dc_{sh}$ and the combined conditions for the row and the column of the applicable entry are satisfied. Thus, s satisfies exactly one of the entries in the first column of Table 10. (Hint: to check this, note that the column headers for T_{lat} of the incidence matrix are equivalent to $s\in t^\bullet$ with $t\neq h$, $s\in{}^\bullet t$ with $t\neq h$, and $t=h$, respectively.)

To learn more about s, we distinguish pre- and post-set of t. If $p\in{}^\bullet t\cap P_h$, then a “$-$”-entry of the incidence matrix applies, which is reflected in one of the upper two rows of Table 10, so that $p\in c$ is obtained from the third column. Similarly, for $p\in t^\bullet\cap P_h$, a “$+$”-entry of the incidence matrix applies, which is reflected in one of the lower four rows of Table 10, hence $p\notin c$. This together yields $(*)$. $\square$

If in a case c the marking on P_s cannot be changed by a τ-sequence we say that c is *emergent on* P_s. The word is meant to hint to the fact that the system must emerge from invisible action to change the marking on P_s. (A corresponding notion cannot be expected for hot expansions since here the tokens can always be shuffled around by τ-transitions, see Lemma 20.) A case c is *emergent* if it is emergent on all P_s, $s\in P_{\text{lat}}$. Formally, emergence is defined by just listing a set of markings. Table 6 shows that

Table 11
Obtaining local emergence on P_s

$c \cap P_s$	w_s	$(c \cap P_s) \triangleright w_s$	
$l(s) > 0$			
$\underline{\{x_s, z_s, r_s\}}$	λ	$\underline{\{x_s, z_s, r_s\}}$	
$\{y_s, z_s, r_s\}$	$q_s t_{zs}$	$\underline{\{x_s, z_s, r_s\}}$	
$\{x_s, c_{zs}, r_s\}$	t_{zs}	$\underline{\{x_s, z_s, r_s\}}$	
$\{y_s, c_{zs}, r_s\}$	$t_{zs} q_s t_{zs}$	$\underline{\{x_s, z_s, r_s\}}$	
$\{c_{xs}, c_{1s}\}$	t_{xs}	$\underline{\{x_s, c_{1s}\}}$	
$\underline{\{x_s, c_{is}\}}$	λ	$\underline{\{x_s, c_{is}\}}$	$1 \leqslant i \leqslant l(s)$
$\{y_s, c_{is}\}$	t_{is}	$\underline{\{x_s, c_{i+1,s}\}}$	$1 \leqslant i < l(s)$
$\{y_s, c_{l(s),s}\}$	$t_{l(s),s}$	$\underline{\{x_s, z_s, w_s\}}$	
$\underline{\{x_s, z_s, w_s\}}$	λ	$\underline{\{x_s, z_s, w_s\}}$	
$\{y_s, z_s, w_s\}$	$q_s t_{zs}$	$\underline{\{x_s, z_s, w_s\}}$	
$\{x_s, c_{zs}, w_s\}$	t_{zs}	$\underline{\{x_s, z_s, w_s\}}$	
$\{y_s, c_{zs}, w_s\}$	$t_{zs} q_s t_{zs}$	$\underline{\{x_s, z_s, w_s\}}$	
$l(s) = 0$			
$\underline{\{r_s\}}$	λ	$\underline{\{r_s\}}$	
$\underline{\{w_s\}}$	λ	$\underline{\{w_s\}}$	

the definition meets our intention, because its rows for τ-transitions do not contain any emergent cases.

$$
\begin{aligned}
c \text{ emergent on } s :\Leftrightarrow\ & c \in C_{\text{impl}} \ \wedge \\
& (s \in P^{+}_{\text{lat}} \Rightarrow c \cap P_s \in \{\underline{\{x_s, z_s, r_s\}}, \underline{\{x_s, z_s, w_s\}}\} \cup \{\underline{\{x_s, c_{is}\}} \mid 1 \leqslant i \leqslant l(s)\}) \ \wedge \\
& (s \in P^{0}_{\text{lat}} \Rightarrow c \cap P_s \in \{\underline{\{r_s\}}, \underline{\{w_s\}}\}). \\
c \text{ emergent } \quad :\Leftrightarrow\ & \forall s \in P_{\text{lat}}\, (c \text{ emergent on } s).
\end{aligned}
$$

In the definition and also in the sequel restrictions of markings to P_s that pertain to emergent cases are underlined. Emergent cases happen to be the same as those for initial cases (by design rather than by necessity).

That an emergent case, even a hot normal emergent case, can always be reached by a τ-sequence is expressed by the following lemma. Its proof is prepared by Table 11 that solves the task for hot normal cases and locally for any given P_s, $s \in P_{\text{lat}}$. In its first column, the table contains the restrictions to P_s of all valid markings (Lemma 4), all the more of cases (Lemma 6). The second column associates to each of them a short τ-sequence $w_s \in T_s^*$ which is enabled in that case. The third column contains the restriction to P_s of the resulting case after firing w_s. That the table is correct is checked row by row with the incidence matrices for place and hot expansion (the latter specifically for $t_{l(s),s}$). By definition of emergence, the resulting cases are emergent on s.

Lemma 24 (Reaching an emergent case).
$\forall c \in C_{\mathrm{impl}}\, \exists\, w \in T_\tau^*$ ($c \triangleright w$ emergent and hot normal).

Proof. A τ-sequence w is explicitly constructed, which is then shown to have the desired property. w will be composed from special sequences which are defined first.

By Lemma 22, we can assume a τ-sequence $w' \in T^*_{\mathrm{hotexp}}$ such that $c' := c \triangleright w'$ is hot normal. For each $s \in P_{\mathrm{lat}}$, we define w''_s to be the τ-sequence provided by Table 11 with the hot normal case c' for c. Thus, each w''_s, $s \in P_{\mathrm{lat}}$ is enabled in c', and the third column of Table 11 tells us that $c' \triangleright w''_s$ is emergent on P_s. From the incidence matrix we see that $vic(w''_s) \subseteq P_s \cup P_{\mathrm{hotexp}}$, so that w''_s does not change any other P_{s_o}, $s_o \in P_{\mathrm{lat}} - \{s\}$. From the incidence matrix for latency place expansion can be seen that transitions from different latency place expansions are independent. The $w''_s \in T^*_s$ are therefore mutually independent. Hence any sequence w'' obtained by stringing together all w''_s, $s \in P_{\mathrm{lat}}$ in arbitrary order is enabled in c' by (Net1). Further, since each w''_s does not change any other P_{s_o}, $s_o \neq s$, the resulting marking $c' \triangleright w''$ is emergent on all P_s, $s \in P_{\mathrm{lat}}$, that is, it is emergent. As $c' \triangleright w'' = c \triangleright w'w''$, the τ-sequence $w'w''$ leads to an emergent case.

By construction, $c' \triangleright w''$ is also hot normal, because any $t_{l(s),s}$ occurring in w'' removes a token from dc_{sh} coincidentally with achieving $\neg\, obstr(s,h,c)$. But rather than proving this, it is easier to append just another τ-sequence w''' according to Lemma 22 to arrive at a hot normal case. This is still emergent since $w''' \in T^*_{\mathrm{hotexp}}$ and hence does not change P_{placeexp}, as is seen from the incidence matrix for hot expansion. The desired τ-sequence then is $w := w'w''w'''$. $\square$

7. Equivalence

Whenever systems are compared with respect to their behaviour, the intriguing question arises as to which equivalence notion should be applied. In net theory, there is a large variety of substantially different equivalence notions, and the criteria for selecting the most relevant ones for a particular situation are by no means obvious. However, in our case, there is some evidence that all known concepts might have the same power of distinction, so that the choice of equivalence would not really matter. We do not have a proof for this, but a detailed proof exists that an implementation satisfies the so-called B-condition for labelled elementary systems (see [8]).[2] Loosely speaking, the B-condition says that an unobservable decision cannot lead to distinct observable future behaviours. More precisely, if two initial firing sequences have

[2] The proof is not contained in this paper because of its size, but it will be made available either as a separate publication, or else as a technical report that can be obtained from the author.

identical associated label sequences, then for each continuation of one of them there exists a continuation of the other one such that both have the same associated label sequences.

In system modelling, the B-condition is a very reasonable requirement when "implementations" are considered. By an implementation we mean a system that has all transitions of a system to be modelled, and additionally any (finite) number of auxiliary transitions, and that generates exactly the same initial firing sequences as the original system if auxiliary transitions are ignored. From this arises, a natural and particularly simple labelling function that is identity on all transitions of the original system and maps auxiliary transitions to the empty string.

The systems Σ_{impl} introduced in Section 5 are designed such that they are implementations in this sense. But ultimately we want to compare also other "reasonable" implementations of a given place latency system with respect to equivalence. The point is, of course, that we suggest to consider an implementation reasonable only if it satisfies the B-condition. With this provision, we can say that all reasonable implementations of a given system are equivalent under *any* of the usually considered equivalence notions. This is a result reported in [8]. (Note that, by definition, all reasonable implementations are string equivalent.)

One might argue about our view of a "reasonable implementation". It would therefore be preferable if we could discuss equivalence issues in direct comparison of the given place latency system Σ_{lat} with its implementation Σ_{impl}. For this, however, we are impeded by the fact that Σ_{lat} is not a Petri net, so that we cannot use results from net theory, but need to look for a more general framework. If the result on the B-condition would analogously be valid for transition systems in general [15], a translation of the equivalence relations between Σ_{lat} and Σ_{impl} to that framework should lead to a corresponding result. The work done in [6] should be helpful for this. We have not pursued this idea. The prospects for success are, however, favourable, since Σ_{lat} is unlabelled (has an identity labelling).

We show now that a place latency system Σ_{lat} and its implementation Σ_{impl} are string equivalent under the labelling function α. (α is defined in Section 6.1.)

We will have to examine in parallel for Σ_{lat} and Σ_{impl} the progression of cases effected by the firing of visible transitions. Table 12 provides a convenient overview of the changes a case c and its associated φ-image $\varphi(c)$ undergoes on firing such a transition. For a given $s \in P_{\text{lat}}$, only those visible transitions are considered that affect s in Σ_{lat}, or, equivalently, P_s in Σ_{impl}, that is, which are from $vic(s) \cup \{\pi\}$.

Table 12 is an extraction of Tables 6 and 5, together with an application of the firing rule for Σ_{lat}. Columns 1, 2 and 5 are obtained from Table 6 by leaving away the rows for τ-transitions. The other columns contain derived values. Column 3 is taken from Table 5, applied to the second column. Column 4 is deduced from the firing rule for Σ_{lat}, applied to the first and third column. Column 6 is taken from Table 5, applied to the fifth column.

The table associates φ-values, transitions and follower markings to valid markings, hence, by Lemma 6, to all cases of Σ_{impl}. The transitions included are only those

Table 12
φ- and $\triangleright$-values of cases in Σ_{impl} and Σ_{lat}

	$c \cap P_s$ (in Σ_{impl})	$\varphi(c)(s)$ (in Σ_{lat})	$(\varphi(c) \triangleright t)(s)$ (in Σ_{lat})	$(c \triangleright t) \cap P_s$ (in Σ_{impl})	$\varphi(c \triangleright t)(s)$ (in Σ_{lat})	
$l(s) > 0$						
$t \in {}^\bullet s$	$\underline{\{x_s, z_s, r_s\}}$	$\circ$	$l(s)$	$\{c_{xs}, c_{1s}\}$	$l(s)$	
$t \in s^\bullet$	$\underline{\{x_s, z_s, r_s\}}$ $\{y_s, z_s, w_s\}$ $\{x_s, c_{zs}, w_s\}$ $\{y_s, c_{zs}, w_s\}$	$\bullet$	$\circ$	$\underline{\{x_s, z_s, r_s\}}$ $\{y_s, z_s, r_s\}$ $\{x_s, c_{zs}, r_s\}$ $\{y_s, c_{zs}, r_s\}$	$\circ$	
$t = \pi$	$\underline{\{x_s, z_s, r_s\}}$ $\{x_s, c_{zs}, r_s\}$	$\circ$	$\circ$	$\{y_s, z_s, r_s\}$ $\{y_s, c_{zs}, r_s\}$	$\circ$	
	$\underline{\{x_s, c_{is}\}}$	$l(s) - i + 1$	$l(s) - i$	$\{y_s, c_{is}\}$	$l(s) - i$	$1 \leqslant i < l(s)$
	$\underline{\{x_s, c_{l(s),s}\}}$	1	$\bullet$	$\underline{\{y_s, c_{l(s),s}\}}$	$\bullet$	
	$\underline{\{x_s, z_s, w_s\}}$ $\{x_s, c_{zs}, w_s\}$	$\bullet$	$\bullet$	$\{y_s, z_s, w_s\}$ $\{y_s, c_{zs}, w_s\}$	$\bullet$	
$l(s) = 0$						
$t \in {}^\bullet s$	$\underline{\{r_s\}}$	$\circ$	$\bullet$	$\underline{\{w_s\}}$	$\bullet$	
$t \in s^\bullet$	$\underline{\{w_s\}}$	$\bullet$	$\circ$	$\underline{\{r_s\}}$	$\circ$	

belonging to both systems, Σ_{impl} and Σ_{lat}, and affecting P_s in Σ_{impl} or, equivalently, s in Σ_{lat}. That is, they are transitions from $vic(s) \cup \{\pi\}$.

More precisely, the table contains in the second column the restrictions to P_s of all valid markings of Σ_{impl} in which some visible transition may be enabled. The remaining columns contain the considered derived values. Column 3 does not even need $c \triangleright t$ since it follows from the second column. Column 4, however, is applicable only if also $\varphi(c) \triangleright t$. Comparison of columns 4 and 6 shows

Lemma 25 (φ and visible transitions commute).

$$\forall s \in P_{\mathrm{lat}},\ t \in vic(s) \cup \{\pi\},\ c \in C_{\mathrm{impl}}\ (\varphi(c) \triangleright t \wedge c \triangleright t \Rightarrow (\varphi(c) \triangleright t)(s) = \varphi(c \triangleright t)(s)).$$

The next lemma says that with a τ-transition one cannot leave a representation class.

Lemma 26 (Representation classes closed wrt. τ-transitions).
$\forall w \in T_\tau^*, \quad c \in C_{\text{impl}}\,(c \triangleright w \Rightarrow \varphi(c) = \varphi(c \triangleright w))$.

Proof. By recursion over the length of w.

The lemma is trivially true for $w = \lambda$. Let $w = w't$ and define $c' := c \triangleright w'$, so that $c \triangleright w = c' \triangleright t$. By the recursion assumption we have $\varphi(c) = \varphi(c \triangleright w') = \varphi(c')$.

For $t \in T_{\text{hotexp}}$, we observe that $\dot{vic}(T_{\text{hotexp}}) \subseteq P_{\text{hotexp}}$ by the incidence matrix for hot expansion. Therefore, the changes effected by t are within P_{hotexp}. Since, by definition, φ does not depend on the marking of P_{hotexp} (see Table 5), we have $\varphi(c') = \varphi(c' \triangleright t)$.

For $t \in T_{\text{placeexp}}$, there is an $s \in P_{\text{lat}}^+$ such that $t \in T_s$. The incidence matrix tells us that t does not affect any $P_{s'}$, $s' \neq s$, so that $c' \cap P_{s'} = c' \triangleright t \cap P_{s'}$ for $s' \in P_{\text{lat}} - \{s\}$. The rows for T_s of Table 6 give the changes on P_s for all cases in which t is enabled. Comparison with Table 5 shows that case and follower case have the same φ-values on s. Since they are identical on all other $P_{s'}$, they are in the same representation class, that is, $\varphi(c') = \varphi(c' \triangleright t)$.

Thus, we have $\varphi(c') = \varphi(c' \triangleright t)$ for all $t \in T_\tau$. Since $\varphi(c) = \varphi(c')$ and $c \triangleright w = c' \triangleright t$, the claimed property follows. $\square$

Lemma 27 (Enabling translated to Σ_{lat}).
$\forall t \in T_v, \quad c \in C_{\text{impl}}(c \triangleright t \Rightarrow \varphi(c) \triangleright t)$.

Proof. According to the definition of "$\triangleright$" in Σ_{lat} we have to show $\varphi(c)({}^\bullet t) = \{\bullet\}$ and $\varphi(c)(t^\bullet) = \{\circ\}$ if $t \in T_{\text{lat}}$, and $\neg \exists h \in H(\varphi(c) \triangleright h)$ if $t = \pi$.

For $c \triangleright t$, Table 12 is applicable. It directly yields $\varphi(c)(s) = \bullet$ for $t \in s^\bullet$ and $\varphi(c)(s) = \circ$ for $t \in {}^\bullet s$, hence $\varphi(c)({}^\bullet t) = \{\bullet\}$ and $\varphi(c)(t^\bullet) = \{\circ\}$.

For $t = \pi$, we use an indirect argument by assuming some $h \in H$ such that $\varphi(c) \triangleright h$. This means $\varphi(c)(s) = \bullet$ for any $s \in {}^\bullet h$. Since $c \triangleright t$, Lemma 5 yields $w_s \in c$. Similarly, $r_s \in c$ for any $s \in h^\bullet$. By the definition of *obstr* follows $\neg\, obstr(s, h, c)$. The definition also yields $\neg\, obstr(s, h, c)$ for any other latency place s, that is, $\|\{s \mid obstr(s, h, c)\}\| = 0$. By Lemma 19 (deficiency count property) we obtain $c \cap P_h = \emptyset$, and, since $f_h \in P_h$, finally $f_h \notin c$. Now the incidence matrix for hot expansion says that $f_h \in {}^\bullet \pi$, so that we arrive at $\neg\, c \triangleright \pi$, contradicting $c \triangleright t$. $\square$

Lemma 27 asserts that a visible transition that is enabled in a case c of Σ_{impl} is also enabled in the modelled case $\varphi(c)$ of Σ_{lat}. The converse, however, is not true. In the representation class for $\varphi(c)$ there may be cases in which even no visible transition is enabled. But at least it is possible to reach by τ-transitions a case in which the given visible transition is enabled. This is expressed in the following lemma.

Lemma 28 (Enabling translated to Σ_{impl}).
$\forall t \in T_v, \quad c \in C_{\text{impl}}(\varphi(c) \triangleright t \Rightarrow \exists w \in T_\tau^*(c \triangleright wt))$.

Table 13
Vicinity in latency place expansion

$s \in$	$\varphi(c_{\rm fin})(s)$	$c_{\rm fin} \cap P_s$	${}^\bullet t \cap P_s$	$t^\bullet \cap P_s$
${}^\bullet t$	$\bullet$	$\underline{\{x_s, z_s, w_s\}}$	$\underline{\{w_s\}}$	$\underline{\{r_s\}}$
$t^\bullet, l(s)>0$	$\circ$	$\underline{\{x_s, z_s, r_s\}}$	$\underline{\{x_s, z_s, r_s\}}$	$\{c_{xs}, c_{1s}\}$
$t^\bullet, l(s)=0$	$\circ$	$\underline{\{r_s\}}$	$\underline{\{r_s\}}$	$\underline{\{w_s\}}$

Proof. The lemma is proven separately for $t \in T_{\rm lat}$ and $t=\pi$. For both, we first proceed from c by $w_{\rm eh} \in T_\tau^*$ to an emergent, hot normal case $c_{\rm eh} := c \triangleright w_{\rm eh}$. Such a sequence $w_{\rm eh}$ exists by Lemma 24. $c_{\rm eh}$ is further continued by a τ-sequence $w_{\rm fin} \in T_{\rm hotexp}^*$, to be defined differently for $t \in T_{\rm lat}$ and $t=\pi$, by which $c_{\rm fin} := c_{\rm eh} \triangleright w_{\rm fin}$ is reached. Lemma 26 [$w_{\rm eh}$ for w] and [$w_{\rm fin}$ for $w; c_{\rm eh}$ for c] yields $\varphi(c)=\varphi(c \triangleright w_{\rm eh})=\varphi(c \triangleright w_{\rm eh} w_{\rm fin})$, or $\varphi(c)=\varphi(c_{\rm eh})=\varphi(c_{\rm fin})$. Hence, since $\varphi(c) \triangleright t$, we have $\varphi(c_{\rm eh}) \triangleright t$ and $\varphi(c_{\rm fin}) \triangleright t$. Because $\dot{vic}(T_{\rm hotexp}) \subseteq P_{\rm hotexp}$, $w_{\rm fin}$ affects $P_{\rm hotexp}$ only. Since emergence is defined irrespective of $P_{\rm hotexp}$, the reached case $c_{\rm fin} = c_{\rm eh} \triangleright w_{\rm fin}$ is still emergent (but not necessarily hot normal).

We are finished if we can prove $c_{\rm fin} \triangleright t$, because $w := w_{\rm eh} w_{\rm fin}$ then has the desired property. For the proof, the pre-set and the post-set of t are decomposed into their place latency expansion and their hot expansion parts (see definition of $P_{\rm impl}$), that is, ${}^\bullet t = \{{}^\bullet t \cap P_s \mid s \in P_{\rm lat}\} \cup \{{}^\bullet t \cap P_h \mid h \in H\}$ and $t^\bullet = \{t^\bullet \cap P_s \mid s \in P_{\rm lat}\} \cup \{t^\bullet \cap P_h \mid h \in H\}$. What then has to be shown is
(A) ${}^\bullet t \cap P_s \subseteq c_{\rm fin}$ and $t^\bullet \cap P_s \cap c_{\rm fin} = \emptyset$ for all $s \in P_{\rm lat}$,
(B) ${}^\bullet t \cap P_h \subseteq c_{\rm fin}$ and $t^\bullet \cap P_h \cap c_{\rm fin} = \emptyset$ for all $h \in H$.

$t \in T_{\rm lat}$

For $t \in T_{\rm lat}$, we set $w_{\rm fin} := \lambda$, thus $c_{\rm eh} = c_{\rm fin}$, so that, like $c_{\rm eh}$, $c_{\rm fin}$ is hot normal. Thus, Lemma 23 says it is sufficient to prove (A).

Let $s \in P_{\rm lat}$. If $s \notin vic(t)$, the incidence matrix for latency place linkage yields $\dot{vic}(t) \cap P_s = \emptyset$, from which trivially follows (A). $s \in vic(t)$ is handled by Table 13. Column 1 covers all of $vic(t)$. Column 2 is a consequence of column 1 and $\varphi(c_{\rm fin}) \triangleright t$. Column 3 follows from Table 5 and $c_{\rm fin}$ emergent. Columns 4 and 5 are directly taken from the incidence matrix for latency place linkage, using column 1. (A) results from column 3 together with columns 4 and 5, respectively.

$t = \pi$

From $\varphi(c_{\rm eh}) \triangleright \pi$ and the definition of π-enabling we know $\forall h \in H\, (\neg \varphi(c_{\rm eh}) \triangleright h)$, from which follows $\forall h \in H (\neg\, c_{\rm eh} \triangleright h)$ by Lemma 27. Lemma 21 then yields a $w_{\rm fin} \in T_{\rm hotexp}^*$ such that $f_h \in c_{\rm eh} \triangleright w_{\rm fin}$ and $c_{fh} \notin c_{\rm eh} \triangleright w_{\rm fin}$, or, more concisely, $f_h \in c_{\rm fin}$ and $c_{fh} \notin c_{\rm fin}$ for all $h \in H$. From this follows (B) by the incidence matrix for hot expansion.

If $l(s)=0$, then, according to the incidence matrix for latency place linkage, $\dot{vic}(\pi) \cap P_s = \emptyset$. Thus (A) holds trivially. Let now $l(s)>0$. Since $c_{\rm fin}$ is emergent, $x_s \in c_{\rm fin}$ and $y_s \notin c_{\rm fin}$ by definition. Therefore, the incidence matrix for latency place linkage yields (A). $\square$

For an overview diagram of the proof see Fig. 7.

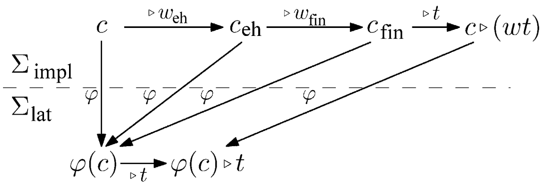


Fig. 7. Overview for proof of Lemma 28.

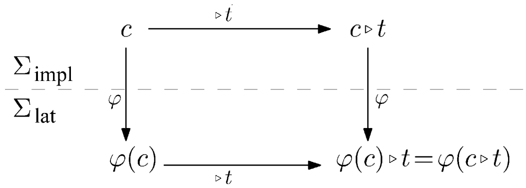


Fig. 8. φ and $\triangleright$ commute for visible transitions.

With Lemma 25, we have established that φ commutes with the firing of visible transitions. For invisible transitions this cannot literally be true, because these do not occur in Σ_{lat}. Rather, they are supposed to simply disappear. The following lemma extends Lemma 25 to τ-transitions and uses the labelling function α for uniformly stating the commutation property for both, visible and invisible transitions.

Lemma 29 (Commuting diagram for φ and $\triangleright t$).

$$\forall t \in T_{\text{impl}}, \quad c \in C_{\text{impl}}(c \triangleright t \Rightarrow \varphi(c) \triangleright \alpha(t) = \varphi(c \triangleright t)).$$

Proof. Let $s \in P_{\text{lat}}$ and $t \in vic(s) \cup \{\pi\}$. From Lemma 25, we have $(\varphi(c) \triangleright t)(s) = \varphi(c \triangleright t)(s)$.

If $t \in T_{\text{lat}} - vic(s)$, then t does not affect P_s, that is, $c \cap P_s = (c \triangleright t) \cap P_s$, and hence $\varphi(c)(s) = \varphi(c \triangleright t)(s)$. Also, t does not affect $s \in P_{\text{lat}}$ in Σ_{lat}, that is, $\varphi(c)(s) = (\varphi(c) \triangleright t)(s)$. Thus, $(\varphi(c) \triangleright t)(s) = \varphi(c \triangleright t)(s)$ holds also for $t \in T_{\text{lat}} - vic(s)$.

Since $s \in P_{\text{lat}}$ was arbitrary, and $(vic(s) \cup \{\pi\}) \cup (T_{\text{lat}} - vic(s)) = T_v$ according to the definition of T_v, this together yields $\varphi(c) \triangleright t = \varphi(c \triangleright t)$ for $t \in T_v$.

The same is not true for $t \in T_\tau$, because then t does not belong to Σ_{lat} and consequently $\varphi(c) \triangleright t$ is meaningless. However, if we take $\alpha(t)$ instead, we obtain $\varphi(c) \triangleright \alpha(t) = \varphi(c \triangleright t)$, which is trivially true because $\alpha(t) = \lambda$, and by Lemma 26 we know already $\varphi(c \triangleright t) = \varphi(c)$.

Since for $t \in T_\tau$ we have $\alpha(t) = \lambda$, the results can be combined into the one formula. $\square$

For $t \in T_v$, the lemma is visualized by the commuting diagram in Fig. 8.

For $t \in T_\tau$, the diagram degenerates to that in Fig. 9

We are now ready to prove that a place latency system is string equivalent with its implementation in the sense that the α-images of the sequences generated by the

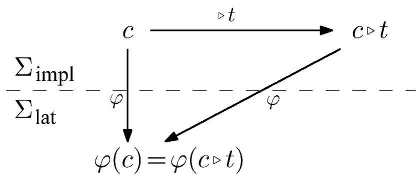


Fig. 9. φ and $\triangleright$ commute for invisible transitions (collapsed).

implementation are exactly the sequences generated by the place latency system. More straightly expressed: the systems generate exactly the same sequences if τ-transitions are ignored. Conceiving the implementation as a model of the place latency system, we show the model to be *correct*, i.e., it does not generate wrong sequences, and *complete*, i.e, it generates *all* necessary sequences.

For a single transition, the correctness issue is already solved by Lemma 29. For a general sequence, it follows by simple recursion.

Lemma 30 (Model correctness).
$\forall w \in T^*_{\text{impl}}(c^\circ_{\text{impl}} \triangleright w \Rightarrow \varphi(c^\circ_{\text{impl}} \triangleright w) = c^\circ_{\text{lat}} \triangleright \alpha(w)).$

Proof. The lemma is trivially true for $w = \lambda$, because $\varphi(c^\circ_{\text{impl}}) = c^\circ_{\text{lat}}$ by Lemma 3. For $w = w't$, the recursion assumption yields $\varphi(c^\circ_{\text{impl}} \triangleright w') = c^\circ_{\text{lat}} \triangleright \alpha(w')$. Lemma 29 [$c^\circ_{\text{impl}} \triangleright w'$ for c] yields $\varphi(c^\circ_{\text{impl}} \triangleright w') \triangleright \alpha(t) = \varphi(c^\circ_{\text{impl}} \triangleright w't)$. This together results in $\varphi(c^\circ_{\text{impl}} \triangleright w) = \varphi(c^\circ_{\text{impl}} \triangleright w't) = \varphi(c^\circ_{\text{impl}} \triangleright w') \triangleright \alpha(t) = (c^\circ_{\text{lat}} \triangleright \alpha(w')) \triangleright \alpha(t) = c^\circ_{\text{lat}} \triangleright \alpha(w't) = c^\circ_{\text{lat}} \triangleright \alpha(w)$. $\square$

Proving model completeness needs more effort, because, in general, τ-transitions have to be inserted in Σ_{impl} for enabling the next visible transition.

Lemma 31 (Model completeness).
$\forall w \in T^*_v(c^\circ_{\text{lat}} \triangleright w \Rightarrow \exists w_{\text{impl}} \in T^*_{\text{impl}}(\varphi(c^\circ_{\text{impl}} \triangleright w_{\text{impl}}) = c^\circ_{\text{lat}} \triangleright w \wedge \alpha(w_{\text{impl}}) = w)).$

Proof. Again, the proof is trivial for $w = \lambda$. $w_{\text{impl}} := \lambda$ then has the desired properties.

Let now $w = w't$ with $w' \in T^*_v$, $t \in T_v$. Note that $\alpha(t) = t$ by the definition of α. The recursion assumption permits to assume a $w'_{\text{impl}} \in T^*_{\text{impl}}$ that solves the issue for w', i.e., $\varphi(c^\circ_{\text{impl}} \triangleright w'_{\text{impl}}) = c^\circ_{\text{lat}} \triangleright w'$ and $\alpha(w'_{\text{impl}}) = w'$. If we abbreviate $c := c^\circ_{\text{impl}} \triangleright w'_{\text{impl}}$, the first equation becomes $\varphi(c) = c^\circ_{\text{lat}} \triangleright w'$. Starting from c, a continuation must be found such that the case reached is mapped by φ to $c^\circ_{\text{lat}} \triangleright w$, and its α-image is t.

Although, by $c^\circ_{\text{lat}} \triangleright w't$, t is enabled in $\varphi(c)$, the same cannot be expected to hold in c, because it may be necessary to insert τ-transitions to make Σ_{impl} ready for firing t. We have already a lemma on translating enabledness in Σ_{lat} back to Σ_{impl}. It guarantees that such τ-transitions exist: Lemma 28 [w_τ for w] provides a $w_\tau \in T^*_\tau$ such that $c \triangleright w_\tau t$. By Lemma 26 [$w_\tau$ for w], w_τ does not change the φ-image, i.e., $\varphi(c \triangleright w_\tau) = \varphi(c)$. Now the diagram for model completeness (see Fig. 10) can be closed using Lemma 29 [$c \triangleright w_\tau$ for c]. It yields $\varphi(c \triangleright w_\tau) \triangleright \alpha(t) = \varphi(c \triangleright w_\tau t)$. With $w_{\text{impl}} := w'_{\text{impl}} w_\tau t$ we obtain $c^\circ_{\text{impl}} \triangleright w_{\text{impl}}$ from $c \triangleright w_\tau t$ and $c = c^\circ_{\text{impl}} \triangleright w'_{\text{impl}}$.

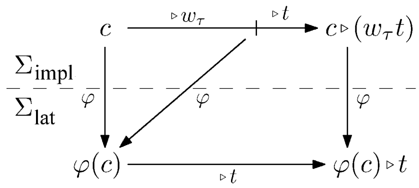


Fig. 10. Overview for proof of Lemma 31.

Combining the partial results finally yields $\varphi(c^{\circ}_{\mathrm{impl}} \triangleright w_{\mathrm{impl}}) = \varphi(c^{\circ}_{\mathrm{impl}} \triangleright w'_{\mathrm{impl}} w_\tau t) = \varphi(c \triangleright w_\tau t) = \varphi(c \triangleright w_\tau) \triangleright \alpha(t) = \varphi(c) \triangleright t = (c^{\circ}_{\mathrm{lat}} \triangleright w') \triangleright t = c^{\circ}_{\mathrm{lat}} \triangleright w$, as required. Further, since $w_\tau \in T_\tau^*$, we also have $\alpha(w_{\mathrm{impl}}) = \alpha(w'_{\mathrm{impl}} w_\tau t) = w't = w$. Thus, w_{impl} has the desired properties. □

From model correctness and completeness follows directly our main result.

Theorem Equ (Equivalence)

$$\{w_{\mathrm{lat}} \mid c^{\circ}_{\mathrm{lat}} \triangleright w_{\mathrm{lat}}\} = \{\alpha(w_{\mathrm{impl}}) \mid c^{\circ}_{\mathrm{impl}} \triangleright w_{\mathrm{impl}}\}.$$

Proof. To show "$\supseteq$", assume $c^{\circ}_{\mathrm{impl}} \triangleright w_{\mathrm{impl}}$. By Lemma 30 [$w_{\mathrm{impl}}$ for w] we obtain $c^{\circ}_{\mathrm{lat}} \triangleright \alpha(w_{\mathrm{impl}})$, which shows that $w_{\mathrm{lat}} := \alpha(w_{\mathrm{impl}})$ is in set on the left-hand side.

Conversely, to show "$\subseteq$", assume $c^{\circ}_{\mathrm{lat}} \triangleright w_{\mathrm{lat}}$, and, by Lemma 31 [$w_{\mathrm{lat}}$ for w], a $w_{\mathrm{impl}} \in T^*_{\mathrm{impl}}$ such that $c^{\circ}_{\mathrm{impl}} \triangleright w_{\mathrm{impl}}$ and $\alpha(w_{\mathrm{impl}}) = w_{\mathrm{lat}}$. Obviously, w_{lat} is in set on the right-hand side. □

Note that Lemmas 30 and 31 are stronger than required for the proof. The equations $\varphi(c^{\circ}_{\mathrm{impl}} \triangleright w) = c^{\circ}_{\mathrm{lat}} \triangleright \alpha(w)$ and $\varphi(c^{\circ}_{\mathrm{impl}} \triangleright w_{\mathrm{impl}}) = c^{\circ}_{\mathrm{lat}} \triangleright w$ in the lemmas provide an explicit translation by φ from Σ_{impl} to Σ_{lat} which is not used for the theorem, but needed for the recursions to go through.

8. Conclusion

From the several suggestions to extend Petri nets to include time, those with token latencies on places (Timed Petri Nets) have been selected as a point of departure for showing that time can be modelled by net-intrinsic means. As a quite general representative of this class, place latency nets have been formally defined together with a firing semantics. A method has been given how to construct for any given place latency system an elementary net system as its implementation. In the implementation, the net-extrinsic "pulse generator" of a place latency system is modelled by an ordinary transition. The implementation was proven to be string equivalent to the original place latency system.

The paper is intended as an initial step of work for showing that net-extrinsic clocks are not necessary *in general* for modelling sequencing aspects of time. If it should turn out that there are other important aspects of time which cannot be modelled on a purely causal basis, it would still be interesting to elaborate the deep reasons for assessing limits of the expressiveness of Petri nets.

The next steps in that work should be to prove implementability for other prototypical time extension approaches. Using the present result, this will be easier if such an approach can be reduced to place latency systems. However, for each approach it is first necessary to establish a formal firing semantics. Usually, the originally provided firing semantics is not formalized and not detailed enough. For example, what happens in a time-out situation with conflicting transitions is normally not explicitly specified. While the answer may be considered obvious, there are other more subtle unclear situations. In particular, a constraint on the maximal time is generally specified after which an enabled transition must fire. The most miniature interruption of enabling resets the time-out count. If a real or rational time base is assumed, the interruption may be arbitrarily short. It is doubtful whether this is actually meant, because it does not seem to have any practical justification. Note that our formalization has avoided this problem by requiring that conflicts involving hot transitions must be resolved before the next pulse occurs. It is expected that the same solution can be employed for formalizing the firing rules of other time extension approaches.

The method of an explicitly represented pulse generator provides a simple way for modelling multi-clock systems. In these, individual clocks can be used to control subsystems, as is typical for all kinds of distributed computing, including circuits on different boards or cooperating workstations. Synchronization of clocks or otherwise handling clock deviations will then become modelling issues. Interesting new questions may turn up in areas of common influence from different clocks, such as interfaces. While it is easy to model maximum deviation requirements by designing appropriate synchronic distances between clocks, clock drift boundaries need some attention to avoid problems with infinite values. Since a bound on the tick ratio of different pulse generators still admits unbounded divergence, its enforcement is not immediately representable by elementary net systems (which are finite by definition). A finitely enforceable specification must be used instead, such as one that involves a finite history only.

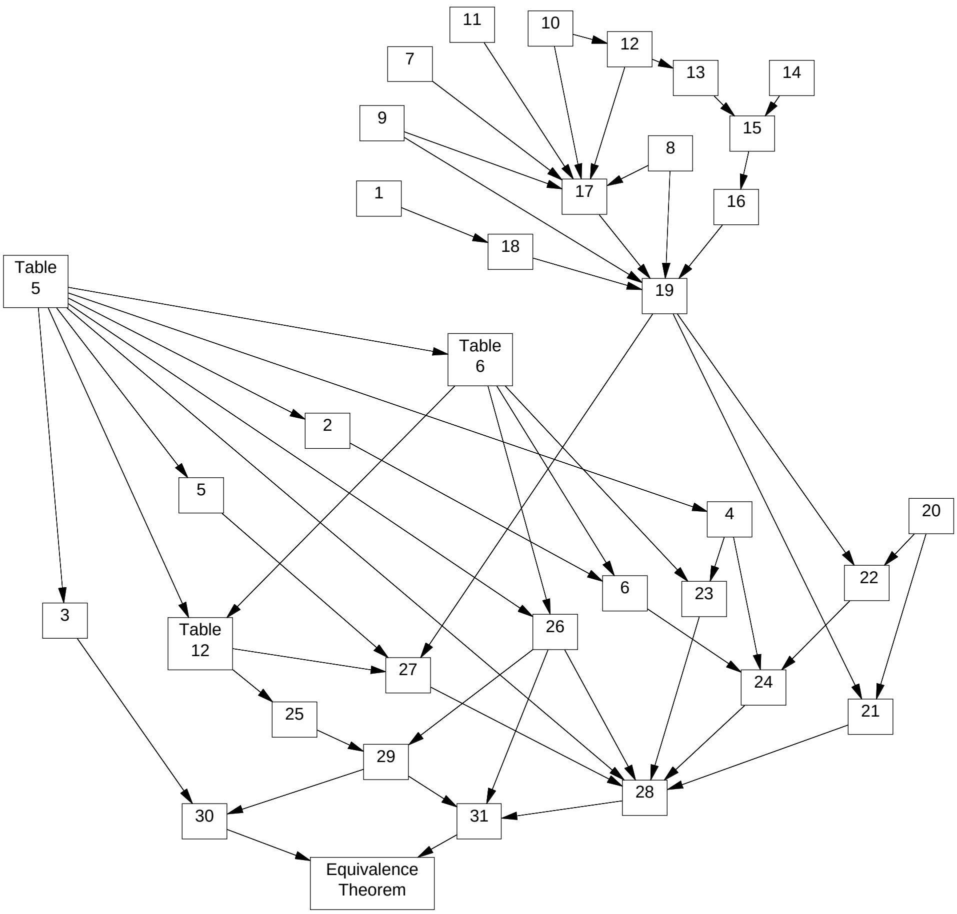
Generally, if one wants to model extended systems with requirements involving infinite values, an implementation with Petri nets is bound to be also infinite. It may be possible to use higher level systems, such as Place/Transition systems with unbounded capacities, for conveniently modelling such extended systems. But if modelling the featured extension needs also additional features in the implementation, this is likely to end up with the power of Turing machines, and thus cannot be reduced to Petri nets proper. This relates also to the above mentioned issue of expressiveness.

Acknowledgements

The idea of representing clock ticks as transitions have since long been around in the Petri Net community. The conviction that the seminal idea should explicitly be stated and backed by formal considerations emerged during an extended period of discussions with Gernot Richter, who took this as a point of departure for developing a practice-oriented scheme for various time-measuring devices. The present theoretical research was conducted in SCAI — Institute for Algorithms and Scientific Computing of GMD — German National Research Center for Information Technology, which provided an environment that made it all possible. In particular, I thankfully acknowledge unending support and encouragement from Thomas Lengauer, as well as interest and advice from Einar Smith. Last, but certainly not least, I owe big thanks to an anonymous referee, whose close scrutiny, detail understanding and useful suggestions greatly contributed to better readability and confidence.

Finally, acknowledgements are due to the computing system, without the help of which I would never have been able to organize the many pieces of drafts in earlier stages during the development of the paper, and, ultimately, the proof itself.

Appendix A. Proof structure



The numbers identify the lemmas. An arrow from A to B means that the proof of B makes use of A. Tables that are defined and used locally only within a proof are not included in the diagram.

Appendix B. Summary of lemmas

Lemma 1 (Criteria for initial case representation).
$\forall s \in P_{\text{lat}}((r_s \in c^{\circ}_{\text{impl}} \Leftrightarrow c^{\circ}_{\text{lat}}(s) = \circ) \wedge (w_s \in c^{\circ}_{\text{impl}} \Leftrightarrow c^{\circ}_{\text{lat}}(s) = \bullet) \wedge (r_s \notin c^{\circ}_{\text{impl}} \vee w_s \notin c^{\circ}_{\text{impl}}))$.

Lemma 2 (Initial case of Σ_{impl} is a valid marking).
c°_{impl} is a valid marking.

Lemma 3 (φ conforms to initial case representation).
$\varphi(c^{\circ}_{\text{impl}}) = c^{\circ}_{\text{lat}}$.

Lemma 4 (Restrictions of valid markings to P_s).

$$\begin{aligned}
&\forall m \subseteq P_{\text{impl}} \ (m \text{ valid marking} \Rightarrow \\
&\forall s \in P^{+}_{\text{lat}}(m \cap P_s \in \{\underline{\{x_s, z_s, r_s\}}, \underline{\{x_s, z_s, w_s\}}, \{x_s, c_{zs}, r_s\}, \{x_s, c_{zs}, w_s\}, \\
&\qquad \{y_s, z_s, r_s\}, \{y_s, z_s, w_s\}, \{y_s, c_{zs}, r_s\}, \{y_s, c_{zs}, w_s\}, \{c_{xs}, c_{1s}\}\} \cup \\
&\qquad \{\underline{\{x_s, c_{is}\}} \mid 1 \leqslant i \leqslant l(s)\} \cup \{\{y_s, c_{is}\} \mid 1 \leqslant i \leqslant l(s)\}) \\
&\wedge \\
&\forall s \in P^{0}_{\text{lat}}(m \cap P_s \in \{\underline{\{r_s\}}, \underline{\{w_s\}}\})).
\end{aligned}$$

Lemma 5 (Characteristic properties of φ-values).
$\forall s \in P_{\text{lat}}, m \subseteq P_{\text{impl}}(m \text{ valid marking} \Rightarrow$
$((\varphi(m)(s) = \circ \Leftrightarrow r_s \in m) \wedge (\varphi(m)(s) = \bullet \Leftrightarrow w_s \in m \vee m = \{y_s, c_{l(s),s}\})))$.

Lemma 6 (Cases of Σ_{impl} are valid markings).
$\forall c \in C_{\text{impl}}(c \text{ is a valid marking})$.

Lemma 7 (Obstruction for enabled hot transition).
$\forall h \in H,\ c \in C_{\text{impl}}\ (c \triangleright h \Rightarrow \{s \mid obstr(s, h, c)\} = \emptyset \wedge \{s \mid obstr(s, h, c \triangleright h)\} = vic(h))$.

Lemma 8 (*readies* and *obstr*).
$\forall h \in H, \quad t \in T_{\text{impl}}, \quad s \in P_{\text{lat}}, \quad c \in C_{\text{impl}}$
$(c \triangleright t \wedge readies(t, s, h) \Rightarrow obstr(s, h, c) \wedge \neg\, obstr(s, h, c \triangleright t))$.

Lemma 9 (*unreadies* and *obstr*).
$\forall h \in H, \quad t \in T_{\text{impl}}, \quad s \in P_{\text{lat}}, \quad c \in C_{\text{impl}}$
$(c \triangleright t \wedge unreadies(t, s, h) \Rightarrow \neg\, obstr(s, h, c) \wedge obstr(s, h, c \triangleright t))$.

Lemma 10 (*readies* and *unreadies* exclude each other).
$\forall h \in H, t \in T_{\text{impl}}, s \in P_{\text{lat}}\ (\neg(readies(t, s, h) \wedge unreadies(t, s, h)))$.

Lemma 11 (Relations between *readies*, *unreadies* and *obstr*).
$\forall t \in T_{\mathrm{impl}}, \quad s \in P_{\mathrm{lat}}, \quad h \in H, \quad c \in C_{\mathrm{impl}}$
$(c \triangleright t \Rightarrow readies(t,s,h) \vee unreadies(t,s,h) \vee (obstr(s,h,c) \Leftrightarrow obstr(s,h,c \triangleright t)))$.

Lemma 12 (Hot transition unreadies its vicinity$_L$ for itself).
$\forall h \in H\,(\{s \mid readies(h,s,h)\} = \emptyset \wedge \{s \mid unreadies(h,s,h)\} = vic(h))$.

Lemma 13 (Characterization of *readies* and *unreadies* in Σ_{impl}).
$\forall h \in H, \quad t \in T_{\mathrm{impl}} - T_{\mathrm{hotexp}}$
$(\{s \mid readies(t,s,h)\} = \{s \mid dc_{sh} \in {}^{\bullet}t\} \wedge \{s \mid unreadies(t,s,h)\} = \{s \mid dc_{sh} \in t^{\bullet}\})$.

Lemma 14 (Correspondence between latency places and representation in P_{hotexp}).
$\forall\, h \in H, \quad P \subseteq P_{\mathrm{impl}}(\|\{s \mid dc_{sh} \in P\}\| = \|\{dc_{sh} \mid dc_{sh} \in P\}\|)$.

Lemma 15 (Correspondence between readying/unreadying and vicinity$_I$ within P_{hotexp})

$\forall h \in H, \quad t \in T_{\mathrm{impl}} - T_{\mathrm{hotexp}} - \{\pi\}$
$(\|\{s \mid readies(t,s,h)\}\| = \|{}^{\bullet}t \cap P_h\| \wedge \|\{s \mid unreadies(t,s,h)\}\| = \|t^{\bullet} \cap P_h\|)$.

Lemma 16 (Cardinality change on hot expansion marking by transition firing).
$\forall h \in H, \quad t \in T_{\mathrm{impl}}, \quad c \in C_{\mathrm{impl}}$
$(c \triangleright t \Rightarrow \|c \triangleright t \cap P_h\| = \|c \cap P_h\| - \|\{s \mid readies(t,s,h)\}\| + \|\{s \mid unreadies(t,s,h)\}\|)$.

Lemma 17 (Obstruction change by transition firing).
$\forall h \in H, \quad t \in T_{\mathrm{impl}}, \quad c \in C_{\mathrm{impl}}$
$(c \triangleright t \Rightarrow \{s \mid obstr(s,h,c \triangleright t)\} =$
$(\{s \mid obstr(s,h,c)\} - \{s \mid readies(t,s,h)\}) \cup \{s \mid unreadies(t,s,h)\})$

Lemma 18 (Deficiency count property for initial case).
$\forall h \in H(\|c^{\circ}_{\mathrm{impl}} \cap P_h\| = \|\{s \mid obstr(s,h,c^{\circ}_{\mathrm{impl}})\}\|)$.

Lemma 19 (Deficiency count property).
$\forall h \in H, \quad c \in C_{\mathrm{impl}}(\|c \cap P_h\| = \|\{s \mid obstr(s,h,c)\}\|)$.

Lemma 20 (Tokens may float freely on a hot expansion).
$\forall m, \quad m' \subseteq P_{\mathrm{impl}}, \quad H' \subseteq H$
$(m \ominus m' \subseteq P_{H'} \wedge \forall h \in H'(\|m \cap P_h\| = \|m' \cap P_h\|) \Rightarrow \exists w \in T^{*}_{H'}(m' = m \triangleright w))$.

Lemma 21 (Enabling π on hot expansion).
$\forall c \in C_{\mathrm{impl}}(\forall h \in H(\neg\, c \triangleright h) \Rightarrow \exists w \in T^{*}_{\mathrm{hotexp}} \forall h \in H(f_h \in c \triangleright w \wedge c_{fh} \notin c \triangleright w))$.

Lemma 22 (Reaching a hot normal case).
$\forall c \in C_{\mathrm{impl}}\, \exists\, w \in T^{*}_{\mathrm{hotexp}}\,(c \triangleright w$ hot-normal$)$.

Lemma 23 (Enabling in hot normal case).
$\forall t \in T_{\mathrm{lat}}, \quad c \in C_{\mathrm{impl}}\,(c$ hot normal $\wedge\, {}^{\bullet}t \cap P_{\mathrm{placeexp}} \subseteq c \wedge t^{\bullet} \cap P_{\mathrm{placeexp}} \cap c = \emptyset \Rightarrow c \triangleright t)$.

Lemma 24 (Reaching an emergent case).
$\forall c \in C_{\mathrm{impl}}\, \exists\, w \in T_\tau^*\ (c \triangleright w$ emergent and hot normal).

Lemma 25 (φ and visible transitions commute).
$\forall s \in P_{\mathrm{lat}},\quad t \in vic(s) \cup \{\pi\},\quad c \in C_{\mathrm{impl}}\ (\varphi(c) \triangleright t \wedge c \triangleright t \Rightarrow (\varphi(c) \triangleright t)(s) = \varphi(c \triangleright t)(s)).$

Lemma 26 (Representation classes closed wrt. τ-transitions).
$\forall w \in T_\tau^*,\quad c \in C_{\mathrm{impl}}\ (c \triangleright w \Rightarrow \varphi(c) = \varphi(c \triangleright w)).$

Lemma 27 (Enabling translated to Σ_{lat}).
$\forall t \in T_v,\quad c \in C_{\mathrm{impl}}\ (c \triangleright t \Rightarrow \varphi(c) \triangleright t).$

Lemma 28 (Enabling translated to Σ_{impl}).
$\forall t \in T_v,\quad c \in C_{\mathrm{impl}}\ (\varphi(c) \triangleright t \Rightarrow \exists w \in T_\tau^*(c \triangleright wt)).$

Lemma 29 (Commuting diagram for φ and $\triangleright t$).
$\forall t \in T_{\mathrm{impl}},\quad c \in C_{\mathrm{impl}}\ (c \triangleright t \Rightarrow \varphi(c) \triangleright \alpha(t) = \varphi(c \triangleright t)).$

Lemma 30 (Model correctness).
$\forall w \in T_{\mathrm{impl}}^*(c_{\mathrm{impl}}^\circ \triangleright w \Rightarrow \varphi(c_{\mathrm{impl}}^\circ \triangleright w) = c_{\mathrm{lat}}^\circ \triangleright \alpha(w)).$

Lemma 31 (Model completeness).
$\forall w \in T_v^*(c_{\mathrm{lat}}^\circ \triangleright w \Rightarrow \exists w_{\mathrm{impl}} \in T_{\mathrm{impl}}^*(\varphi(c_{\mathrm{impl}}^\circ \triangleright w_{\mathrm{impl}}) = c_{\mathrm{lat}}^\circ \triangleright w \wedge \alpha(w_{\mathrm{impl}}) = w)).$

References

[1] B. Berthomieu, M. Menasche, An enumerative approach for analyzing time Petri nets, in: R.E.A. Mason (Ed.), Information Processing 83, Proc. IFIP 9th World Computer Congress, North Holland, Amsterdam, 1983, pp. 41–46.

[2] J.E. Coolahan, N. Roussopoulos, Timing requirements for time-driven systems using augmented Petri nets, IEEE Trans. Software Eng. SE-9 (5) (1983) 603–616.

[3] R. Durchholz, Causality, time, and deadlines, Data Knowledge Eng. 6 (1991) 469–477.

[4] R. Durchholz, Latency time modelling with elementary net systems, Internat. Workshop on Discrete Event Systems, WODES'96, IEEE, Edinburgh, UK, 1996, pp. 82–87.

[5] C. Ghezzi, D. Mandrioli, S. Morasca, M. Pezzè, A general way to put time in Petri nets, 5th Internat. Workshop on Software Specification and Design, May 19–20, 1989, Pittsburgh, ACM SIGSOFT Engineering Notes, Vol.14,3, IEEE Computer Society Press, 1989, pp. 60–67.

[6] R.J. van Glabbeek, The linear time — branching time spectrum, in: J.C.M. Baeten, J.W. Clop (Eds.), CONCUR'90, Lecture Notes in Computer Science, Vol. 458, Springer, Berlin, 1990, pp. 278–297.

[7] P.M. Merlin, D.J. Farber, Recoverability of communication protocols — implications of a theoretical study, IEEE Trans. Software Eng. COM-24 (9) (1976) 1036–1043.

[8] L. Pomello, G. Rozenberg, C. Simone, A survey of equivalence notions for net based systems, in: G. Rozenberg (Ed.), Advances in Petri Nets 1992, Lecture Notes in Computer Science, Vol. 609, Springer, Berlin, 1992, pp. 410–472.

[9] C. Ramchandani, Analysis of asynchronous concurrent systems by Petri nets, Report MAC TR-120, Cambridge MA, 1974.

[10] G. Richter, Clocks and their use for time modeling, in: A. Sernadas, J. Bubenko, Jr., A. Olivé (Eds.), Information Systems: Theoretical and Formal Aspects, Proc. of the IFIP 8.1 Working Conference

on Theoretical and Formal Aspects of Information Systems, Sitges, Barcelona, 16–18 April, 1985, North-Holland, Amsterdam, 1985, pp. 49–66.

[11] G. Rozenberg, P.S. Thiagarajan, Petri nets: basic notions, structure, behaviour, in: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (Eds.), Current Trends in Concurrency, Overviews and Tutorials, Lecture Notes in Computer Science, Vol. 224, Springer, Berlin, 1986, pp. 585–667.

[12] M. Shadad, R. Lipsett, E. Marschner, K. Sheehan, H. Cohen, R. Waxman, D. Ackley, VHSIC hardware description language, Computer 18(2) (1985) 94–103.

[13] J. Sifakis, Use of Petri nets for performance evaluation, in: H. Beilner, E. Gelenbe (Eds.), Measuring, Modelling and Evaluating Computer Systems, Proc. 3rd Internat. Symp. Bonn-Bad Godesberg, October, 3–5, North-Holland, Amsterdam, 1977, pp. 75–93.

[14] E. Smith, On the border of causality: contact and confusion, Theoret. Comput. Sci. 153 (1996) 245–270.

[15] P.S. Thiagarajan, Elementary net systems, in: W. Brauer, W. Reisig, G. Rozenberg (Eds.), Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Lecture Notes in Computer Science, Vol. 254, Springer, Berlin, 1987, pp. 26–59.

[16] W.M. Zuberek, Timed Petri nets and preliminary performance evaluation, 7th Ann. Symp. on Computer Architecture, May 6–8, 1980, IEEE, ACM, La Baule, France, 1980, pp. 88–96.